Intelligent Augmented Reality Training for Assembly and Maintenance

by Giles Westerfield

A thesis submitted in partial fulfilment of the
requirements for the Degree
of Master of Science in Computer Science
University of Canterbury
2012

This document is dedicated to my parents, Monte and Kay Westerfield, who have provided unconditional love and support in all aspects of life.

Acknowledgements

The author wishes to express sincere appreciation to Professors Mitrovic and Billinghurst for their excellent supervision throughout the thesis project. In addition, the author would like to acknowledge the Human Interface Technology Laboratory NZ for its considerable support and expertise with Augmented Reality and Human-Computer Interaction. Special thanks also to the Intelligent Computer Tutoring Group for their assistance in the realm of Intelligent Tutoring Systems and for the use of their ASPIRE authoring system.

Abstract

Augmented Reality can visually convey abstract concepts and 3D spatial information in context with real-world objects, which makes it an ideal tool for training and educational purposes. This masters thesis investigates the use of Augmented Reality to assist with training for manual assembly and maintenance tasks. Improving on prior research, this approach combines Augmented Reality with a robust Intelligent Tutoring System to provide a more effective learning experience. After developing a modular software framework, a prototype was created that teaches the user to assemble hardware components on a computer motherboard. A thorough evaluation of the prototype found that the new intelligent approach significantly improves the learning outcome over traditional Augmented Reality training methods that do not employ Intelligent Tutoring Systems.

Contents

List of Tables

List of Figures

Glossary

Augmented Reality (AR) - A technology that seamlessly blends real-time images of virtual objects with images of the real world such that each virtual object appears to exist at a fixed point in reality.

Domain Model - Part of an Intelligent Tutoring System that represents the expert knowledge to be taught to the student.

Head-Mounted Display (HMD) - A mobile display device worn on the head that provides an immersive first-person perspective.

Head-Up Display (HUD) - A screen-aligned graphical overlay that is not spatially registered within a scene.

Intelligent Tutoring System (ITS) - A computer program that provides customized instruction to the user.

Pedagogical Model - Part of an Intelligent Tutoring System that determines the teaching strategies used to impart the domain knowledge upon the student.

Registration - The act of aligning coordinate systems between real and virtual environments. In particular, image registration refers to the act of inferring 3D spatial coordinates from 2D camera images. A virtual object is considered spatially-registered if it is anchored to a fixed position in the real world.

Scaffolding - A teaching strategy that entails scaling the level of assistance with a task as needed based on how the student performs.

Student Model - Part of an Intelligent Tutoring System that represents the student's current knowledge within the learning domain as he or she progresses through the tutoring process.

Tracker - A system of software and hardware components used to infer the relative position and/or orientation of real objects.

Chapter 1

Introduction

Augmented or Mixed Reality allows the user's view of reality to be combined with virtual content that appears to be spatially registered in the real world (Azuma, 1997). This relatively new technology provides unique opportunities to extend the breadth of human potential across multiple disciplines, including Medicine, Architecture, Navigation and Education.

One area of particular interest is the use of Augmented Reality (AR) to assist with training for manual assembly and maintenance tasks. Whether a person is putting together furniture or repairing a car engine, these types of tasks are inherently spatial in nature, and it can be difficult to teach someone to perform particularly complex manual sequences without close instruction and supervision. Unfortunately, personalized human assistance is not always available or cost effective. Many systems include instruction manuals containing diagrams that detail the necessary steps to be performed, but these can be difficult and time consuming to interpret and transform into applicable knowledge. Video tutorials are often more effective tools because they harness the power of visual instruction, but in many cases the user must repeatedly switch between the contexts of the video and the real world environment. Furthermore, because the content of the video is static, there may be differences between the task described in the video and the real task to be performed.

Augmented Reality technology superimposes visual stimuli on top of the user's view of the real world. It has the potential to revolutionize education and training due to its unique ability to visually convey abstract concepts and 3D spatial infor-

Figure 1.1: Wikitude Drive - AR navigation application for the Android mobile operating system.

mation in context with real world objects. For example, through the use of AR, biology students learning about photosynthesis could walk outside and observe real trees to see a visual model of sunlight being absorbed by proteins containing chlorophyll. Chemistry students could physically assemble virtual atoms into molecules, and astronomy students could hold an interactive 3D model of the solar system in their hands.

In addition to assisting with scholastic education and the teaching of abstract concepts, Augmented Reality can also help with real world tasks. Civil engineers can use AR to see pipes or cables beneath the ground in order to plan construction projects and to know where to dig when they are at the site (Schall et al., 2009). Augmented Reality windshield displays can be incorporated into vehicles to highlight the edge of the road and identify hazards at night and in heavy fog where the driver has difficulty seeing (Tonnis et al., 2005). Navigation is another large application domain, where AR displays can direct users to their destinations or provide additional information about landmarks (Narzt et al., 2006). Figure *1.1* demonstrates Wikitude Drive[1], which is an AR navigation application for the Android mobile operating system.

Augmented Reality can also help with training. Like the aforementioned educational applications, the goal of AR training is also to foster learning, but it

[1]Wikitude Drive - wikitude.com/category/02_wikitude/wikitude-drive, retrieved 7/02/2012

focuses more on teaching the user to perform a practical task as opposed to conveying abstract knowledge. Many would agree that it is best to "learn by doing", and AR has the unique capacity to deliver hands-on training experiences where users receive visual instructions in context with the real world objects they are manipulating. Instead of reading text in a paper manual and following 2D diagrams, a person could simply look at a car engine while the AR display shows the parts that need to be adjusted and the sequence of steps required to replace the alternator. The AR approach has the potential to provide a more intuitive, interactive and efficient training experience, and could open up new possibilities for rapid skill development and personal growth. In theory the same hardware can be used for multiple training applications, and one can imagine a future where people can freely obtain AR tutorials to build or repair nearly anything.

There have been numerous investigations exploring the use of Augmented Reality to assist with training for manual assembly and maintenance. These studies have largely involved procedural tasks where the user follows visual cues to perform a linear series of steps (Feiner and Henderson, 2009; Baird and Barfield, 1999; Caudell and Mizell, 1992). The prior research typically focuses on measuring and maximizing the user's efficiency while using the AR system, rather than concentrating on learning and retention of the underlying skills and principles. To that end, the existing prototypes tend to be static in how they deliver the training content—the experience is the same for every user, and the systems have little regard for whether learning is actually taking place.

Intelligent Tutoring Systems are computer programs that provide customized instruction or feedback to the student while performing a task (Psotka and Mutter, 1988). Using robust Intelligent Tutoring techniques, the goal of this project is to create a more interactive Augmented Reality training experience that reacts uniquely to each user. In addition to providing instructions in the form of 3D models and animations, the tutor actively observes the student's behavior in order to correct mistakes and provide meaningful feedback. By maintaining and updating a model of the student's cognition, the system can focus on filling gaps in preexisting knowledge and improving skills in areas where the student does not perform well. The hope is that this leads to enhanced knowledge retention and a more robust training experience. After all, the goal of Augmented Reality training should be to efficiently impart the necessary skills and reduce reliance on the AR tutor as quickly as possible—not simply to elevate task performance while using the system.

Thesis Summary

This masters thesis began with a thorough review of existing literature in the areas of Intelligent Tutoring Systems and Augmented Reality for education and training. While there has been significant progress in each of these fields, there has been minimal investigation into the combination of techniques from both domains. After conducting the background review, a series of high-level questions were formulated to provide the motivation for the new thesis research. The primary goal was to determine whether the integration of ITS techniques can significantly improve the effectiveness of AR training for assembly and maintenance tasks. To achieve this goal, an AR framework was created that utilizes an ITS to provide a robust and customized learning experience for each user.

To demonstrate the framework, a prototype application was created that teaches users how to assemble hardware components on a computer motherboard. Due to a lack of prior research investigating the combination of the two fields, the first task was to determine the key properties any ITS should have in order to work well with an AR interface in order to teach a physical assembly task. After outlining a list of desired properties, seven existing ITS authoring solutions were examined with respect to these characteristics. The clear winner was ASPIRE, which is the constraint-based authoring system developed by the Intelligent Computer Tutoring Group at the University of Canterbury (Mitrovic et al., 2008). The next task was to use the authoring interface in ASPIRE to create the ITS back-end that controls the training process. After the ITS was completed, the software and hardware components of the AR interface were developed, and the two modules were connected to create a working system.

To evaluate the intelligent AR training approach, a traditional system was created for comparison that was identical in every way except for features relating directly to the ITS. The results of the evaluation revealed that participants who used the intelligent system surpassed those who used the traditional system by an average of 25% on written tests and also significantly outperformed them on physical tests of their knowledge. On average, the intelligent tutor group completed the physical test 30% faster than the traditional group while also making fewer mistakes. These results support the conclusion that the combination of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks can significantly improve the learning outcome over traditional AR approaches that do not employ ITSs. Furthermore, the usefulness of the ITS appears to be directly related to the complexity of the task. For very simple tasks, the student is less likely to make a mistake, and thus the ITS does not greatly influence the learning process. However, for more complex or open-ended tasks, the student makes more mistakes and the robust scaffolding and feedback provided by the ITS has a greater impact.

Chapter 2

Background

2.1 Augmented Reality

Augmented Reality generally refers to technology that overlays virtual images onto real-world objects, registered such that the virtual content appears to exist at a fixed location in the real world. For example, rather than looking at a 3D model of a building on a computer monitor, an architect could use a head-mounted display (HMD) to view the virtual building sitting on top of a real table as if the model actually existed in reality. As the person moves around the table in order to view the building from different angles, the rendered image of the virtual object is updated to show the correct perspective, maintaining the illusion that the object exists at a fixed point in space.

Like many budding technologies, Augmented Reality was first conceived for military applications. AR systems were used in the 1960s for fighter jet cockpits, where they provided pilots with information about aircraft systems, navigation waypoints, hazards and enemy targets. Researchers found they were able to reproduce the information represented by the traditional gauges and screens on the aircraft in graphical displays that superimposed information on top of the pilots' view as they looked at the real world outside. The new displays provided numerous benefits that significantly increased the proficiency of pilots in the field. Flight data could be relayed in a more intuitive, graphical and dynamic manner, and the cognitive processing required to decode the information was reduced. The connection between pilot and aircraft was greatly enhanced, while overall situational and combat awareness also improved (Furness, 1986).

Major developments in consumer hardware and computer vision led to the first academic Augmented Reality prototypes in the 1990s. A number of studies explored the AR domain during this time, and in 1997 R. Azuma conducted a landmark survey of prior research and coined the widely accepted definition of AR: Augmented Reality systems combine real and virtual, are interactive in real time and are spatially registered in 3D (Azuma, 1997). The following year, J. Rekimoto made an important breakthrough with his "Matrix" method for object identification and registration (Rekimoto, 1998). This was the first major implementation of a pose-estimation system utilizing 2D square-shaped bar code patterns that encode unique identification information—otherwise known as marker tracking. H. Kato and M. Billinghurst developed the ARToolkit tracking library in 1999, which further refined Rekimoto's approach (Kato and Billinghurst, 1999). ARToolkit was soon released to the public as an open-source project and became widely popular among Augmented Reality researchers, commercial ventures and private individuals alike. This dramatically increased the exposure and popularity of AR and prompted the exploration of new interdisciplinary applications. ARToolkit and other tracking libraries have more recently been implemented for mobile and web platforms, which has created new avenues for research and commercial adoption such as outdoor AR and web-based AR marketing. The Augmented Reality domain has since grown exponentially and is earning its place as a prominent frontier of modern technological development.

2.1.1 Augmented Reality and Learning

One of the major interdisciplinary applications of Augmented Reality lies in the realm of education. There has been a lot of research investigating the use of AR as a teaching tool to improve learning performance on a range of subjects including Physics (Buchanan et al., 2008), Medicine (Lamounier et al., 2010; Fuchs et al., 1998), Chemistry (Fjeld and Voegtli, 2002), Biology (Weghorst, 2003) and reading skills (Dünser, 2008). One of the first successful studies was conducted at the University of Washington in 2002, where researchers created an AR interface to teach concepts involving the relationship between the earth and sun to undergraduate geography students (Shelton and Hedley, 2002). The participants were shown a virtual 3D model of the earth and sun that they could pick up and manipulate with their hands. The model showed four instances of the earth positioned around the sun at locations which corresponded to the summer and winter solstices and the spring and autumn equinoxes. The students could see how the earth's axis is always tilted in the same direction as it rotates around the sun, resulting in varying amounts of light reaching the northern and southern hemispheres at different times of year. This naturally led to a discussion between researchers and subjects regarding the cause of the seasons.

The AR interface allowed the students to view the model from any angle, and the combination of the virtual view with reality allowed the students and teachers to gesture at features of the model with their hands. The subjects were given a pre-test and a post-test surrounding the AR tutoring session, and the results demonstrated a dramatic improvement in understanding as well as a significant reduction in student misconceptions when compared to traditional (non-AR) teaching methods. The researchers concluded that AR is useful for conveying concepts involving 3D spatial configurations, and that the ability for the student to control the interaction in an intuitive manner provided a strong benefit (Shelton and Hedley, 2002).

One major advantage of the use of Augmented Reality for learning is that it can accommodate a variety of learning styles. One popular scheme for describing learning aptitude was proposed by Felderman and Silverman (Felderman and Silverman, 1988). In a similar fashion to the popular Myers-Briggs personality type classification, Felderman and Silverman found that students can generally be categorized along five continua, which are summarized in Table 2.1 on the following page. Traditional classroom approaches to education generally support students who are passive, intuitive, symbolic, and deductive. Through the use of interactive 3D spatial graphics, AR has the potential to engage students who are active, sensing, visual, and inductive—making it an excellent complement to written textbooks and verbal lecture material. Many AR classroom studies have noted that the most dramatic improvements were seen among the struggling C and D-level students, who were not learning well with the traditional teaching methods, but

Table 2.1: Felderman & Silverman's Learning Continuum (Felderman and Silverman, 1988)

Active/Passive	Actively exploring the world rather than simply experiencing it
Sensing/Intuitive	Feeling the world rather than thinking about it
Verbal/Visual	Also symbolic / visual. Learning through symbolic descriptions of subject matter rather than visual experience
Sequential/Global	Following a process in pieces rather than initially grasping the whole
Inductive/Deductive	Generalizing rule from many examples rather than extrapolating rule from axioms and previously known rules

who in many cases were brought up to the level of A students through the use of interactive AR experiences (Dünser, 2008).

2.1.2 Augmented Reality for Assembly and Maintenance

Training for manual assembly and maintenance is one type of learning that can benefit significantly from the use of AR because these "hands-on" tasks are inherently spatial and lend themselves naturally to visual instruction. Augmented Reality has a significant advantage over other display technologies in this area as a result of its ability to provide 3D animations, text and graphical cues in context with the real objects to be manipulated. Why look at a diagram of a car engine in order to find a particular component when one can can simply look at the real engine and allow an AR display to point it out?

There has been much prior research exploring the use of Augmented Reality for manual tasks. In the industrial assembly realm, P. Caudell and D. Mizell at Boeing developed one of the first landmark prototypes, which assisted with assembling aircraft wire bundles (Caudell and Mizell, 1992). Their goal was to improve worker efficiency and lower costs by reducing reliance on the traditional templates, formboard diagrams and masking devices normally employed in the assembly process. The display used simple wire-frame vector graphics to show the

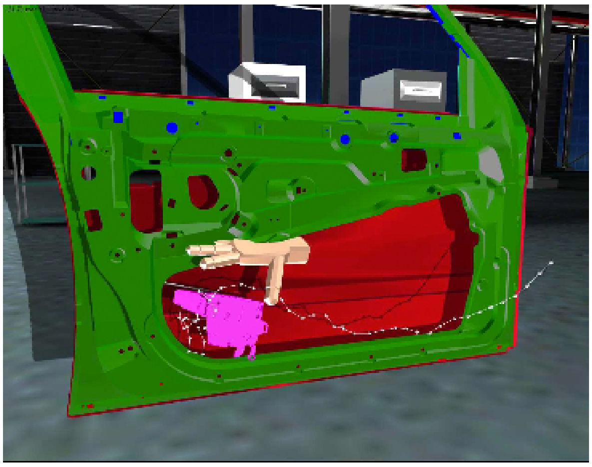


Figure 2.1: Augmented view of a BMW car door and locking mechanism (Reiners et al., 1999).

path of the cable to be added to the bundle, and thus did not employ 3D graphics hardware. The system was later evaluated by D. Curtis et al., who raised a variety of practical concerns regarding the prototype and its deployment in a real aircraft factory (Curtis et al., 1999). Another early investigation involved the creation of a head-mounted monocular AR display to assist with car door lock assembly (Reiners et al., 1999). Figure 2.1 shows the augmented view of a BMW car door locking mechanism. This newer system used 3D CAD models of the car door and internal locking mechanism, which provided improved spatial awareness over the 2D graphics in the aforementioned Boeing wire bundle system. The system guided users through the assembly process in a step-by-step fashion, responding to voice commands to move between the steps. In their evaluation, Reiners et al. found that the prototype was not stable enough for completely new users, whose actions were often not within the anticipated operating parameters. The researchers concluded that some introductory training was required to gain any tangible benefit from the AR system.

These early studies soon led to the formation of several research groups dedicated to exploring the use of AR for industrial applications. ARVIKA was a group based in Germany whose mission was to use AR to implement user-oriented and application-driven support for working procedures in the development, production, and servicing of complex technical products and systems (Friedrich, 2002). They received funding from the German Federal Ministry of Education and Research as well as various corporate sponsors to explore a number of areas, including automobile and aircraft manufacturing and power plant servicing. Their focus was on practicality and applicability, since most previous AR prototypes were too unwieldy to be integrated successfully into industrial workplaces. The researchers conducted usability tests to evaluate ergonomic aspects of AR hardware and software, the time-cost and quality effects of the use of AR in the work process, and the benefit of AR telepresence, which allows specialists to provide remote assistance to technicians in the field. The studies found that the use of AR in industrial development contexts can be extremely beneficial, and that the expensive nature of AR systems is often offset by reduced development time and improved product quality. For example, design engineers were able to rapidly evaluate ergonomic aspects of different aircraft cockpit prototypes by overlaying virtual layout elements over real cockpit mockups, which significantly streamlined the design process. The successful completion of the ARVIKA project in 2003 gave way to the formation of ARTESAS—another German group with renewed funding for research into industrial AR applications (ARTESAS, 2003).

Another related research group, Services and Training through Augmented Reality (STAR), was formed between research institutes in the USA and Europe around the same time (Raczynski and Gussmann, 2004). The primary focus of STAR was to develop new AR techniques for training, documentation and planning purposes. One of the resulting prototypes allows a technician to capture video of the work environment and transmit the images to an off-site specialist. The specialist then annotates the video with drawing and text, which appears in the worker's augmented view in a spatially registered fashion. The researchers found that this method of remote collaboration was an effective means of communicating physical procedures and that it allowed a person with expertise to share his or her knowledge efficiently with multiple trainees in different locations.

A more recent and noteworthy study in the industrial realm was conducted in 2009 by S. Feiner and S. Henderson (Feiner and Henderson, 2009). They developed an Augmented Reality application to support military mechanics conducting routine maintenance tasks inside an armored vehicle turret. In their user study involving real military mechanics, they found that the use of Augmented Reality allowed the subjects to locate components more quickly than when using traditional untracked head-up displays (HUDs) or computer monitors. They also discovered that in some cases the AR condition resulted in less overall head movement, which suggested that it was physically more efficient. The evaluation also included a qualitative survey, which demonstrated that the participants found the Augmented Reality condition to be intuitive and satisfying for the tested sequence of tasks.

In addition to large industrial and manufacturing applications, Augmented Reality has been used to assist with assembly and maintenance on a smaller scale. A study conducted by A. Tang et al. prompted test subjects to assemble toy blocks into specific configurations using several different forms of instruction (Tang et al., 2003). The test conditions included the use of traditional printed media, instructions displayed on an LCD monitor, static instructions displayed via a see-through HMD, and spatially-registered AR instructions also using a HMD. The researchers found that AR instructions overlaid in 3D on top of the toy blocks resulted in an 82% reduction in the error rate for the assembly task. They also found that the AR approach was particularly useful for diminishing cumulative errors, i.e. errors resulting from previous assembly mistakes. Another study by Robertson et al. used a similar set of test conditions and found that subjects assembled toy blocks more quickly using 3D registered AR than with 2D non-registered AR and graphics displayed on a HUD (Robertson et al., 2008).

These toy block assembly studies provide valuable insight, but the tasks performed are somewhat abstract in nature. Augmented Reality has also been applied to real-world assembly tasks in non-industrial settings. One such study conducted by K. Baird and W. Barfield involved the assembly of components on a computer motherboard (Baird and Barfield, 1999). In a similar fashion to the aforementioned toy block studies, the participants were asked to perform the task using a number of different instructional media, which included printed material, slides presented on a computer monitor, and screen-fixed textual instructions on opaque and see-through HMDs. The researchers observed that the test subjects completed the assembly task significantly more quickly and with fewer errors while using the HMD displays. This motherboard assembly task is similar to the one designed for this thesis project, but it should be noted that Baird and Barfield's system did not employ spatially-registered AR and did not utilize Artificial Intelligence when it prompted users to follow a rigid series of assembly steps. In addition, their evaluation concerned itself only with the performance of users while using the display and did not test knowledge retention after the training was complete.

A number of other similar studies have demonstrated positive results for the integration of AR with real-world assembly and maintenance tasks in various domains, including furniture assembly (Zauner et al., 2003), medical assembly (Nilsson and Johansson, 2007) and laser printer maintenance (Feiner et al., 1993).

2.2 Intelligent Tutoring

The rise in information technology in the 20th century generated profound changes in society, but had a relatively low impact in the area of education when compared with many other disciplines. Early technologies such as movies, radio and television were once expected to revolutionize education, but in practice had a limited impact because they generally automated or replicated existing strategies as opposed to fundamentally improving teaching methods (McArthur and Lewis, 1993). Beginning in the 1990s, advances in Artificial Intelligence, Cognitive Science and the Internet have allowed technology to play a more significant role in teaching, where it has since moved beyond mere duplication of traditional strategies.

Intelligent Tutoring Systems (ITSs) are computer programs that provide customized instruction or feedback to the user (student) while performing a task (Psotka and Mutter, 1988). They have surpassed previous e-learning approaches by maintaining models of student knowledge, expert (domain) knowledge, and pedagogical communication. These models allow ITSs to adapt uniquely to each user, focus on problem areas and provide highly detailed feedback to promote learning.

2.2.1 Model-tracing Tutors

Model tracing is the most widely used approach to developing Intelligent Tutoring Systems. Model-tracing tutors rely on production rules, which serve as a collection of strategies for solving problems in a particular domain (Mitrovic et al., 2007). The relevant declarative knowledge about the domain is collected and production rules are formulated to apply the knowledge to a particular problem. An example of declarative knowledge in the domain of geometry is *"If two sides and the included angles of two triangles are congruent, then the triangles are congruent."* Procedural knowledge on the other hand—formalized via production rules—might involve skills of placing triangles into correspondence, determining what an included angle is, setting sub-goals, and making inferences (Anderson and Corbett, 1995). The act of learning takes place when the student comprehends the declarative concept and applies the concept to solve problems through the use of production rules.

There are numerous examples of model-tracing tutoring systems. Researchers at the Learning Research and Development Center at the University of Pittsburgh developed Andes, an intelligent tutor designed to teach physics (Shelby, 2001). Andes was tested for two semesters in a basic physics course taught at the U. S. Naval Academy and yielded positive results as measured by exam scores. J. Anderson et al. of Carnegie-Mellon University developed several tutoring systems to teach concepts in the domains of LISP programming, geometry and algebra (Anderson and Corbett, 1995). These systems were widely successful and were integrated into a number of high school curricula. The algebra tutor, called PAT, was tested in three Pittsburgh high schools in 9th grade algebra classes. On average, the 470 students in the experimental classes outperformed students in regular classes by 15% on standardized tests and by 100% on tests targeting concepts directly covered by the tutor (Koedinger et al., 1997). These and other similar studies demonstrated that laboratory tutoring systems can be robust enough to be deployed in practical and unforgiving settings like metropolitan high schools.

2.2.2 Constraint-based Tutors

Continued progress in Intelligent Tutoring research in the late 1990s gave way to a new approach called constraint-based modeling. Constraint-based tutors represent knowledge in the form of constraints, which state what ought to be so, rather than generating specific problem-solving paths. By distilling domain knowledge into constraints, the tutor can model abstract features of correct solutions, as opposed to generating procedures for performing domain-related tasks the way it is done in model tracing with production rules. Constraints support evaluation and judgment, not inference, and are used to represent both domain and student knowledge (Mitrovic et al., 2007). A sample constraint for the domain of cooking might be *"When making steak, the meat must be allowed to reach room temperature before it is cooked."* The constraint implicitly contains a *relevance condition* that describes when the constraint applies (when cooking steak), a *satisfaction condition* that tests whether the constraint is met (the meat is at room temperature), and a *feedback message* that tells the user what he or she is doing wrong and encourages correct behavior, e.g. *"When preparing a steak, the meat needs to be at room temperature before you can cook it."* By modeling this knowledge as a general constraint instead of a production rule, it can apply to many different scenarios involving steak, as opposed to serving as a specific procedure for making a steak.

One of the primary centers of constraint-based ITS research is the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury. SQL-Tutor, the first constraint-based ITS, taught students how to query relational databases using the SQL language (Mitrovic and Ohlsson, 1999). Through the use of almost 700 constraints, the system supports a wide range of problem types and can provide extremely detailed feedback. The tutor also selects the problems to be presented to the user based on performance with previous problems. The initial user study found that after only two hours with the SQL-Tutor, students outperformed their peers in a post-examination, scoring an average of three quarters of a standard deviation higher on questions related to SQL query formulation (Mitrovic et al., 2007). Following the success of SQL-Tutor, the ICTG has developed a number of ITSs for a range of domains, including enhanced entity-relationship modeling, English grammar and punctuation, data normalization and arithmetic relating to capital investment.

2.2.3 ITS Authoring

A major drawback to Intelligent Tutoring Systems is that they are often quite complex and time-consuming to create. For example, it took approximately one hour to fully develop and test each of the nearly 700 constraints in SQL-Tutor (Mitrovic and Ohlsson, 1999). Furthermore, this metric is for an ITS expert who is familiar with the constraint syntax and who is experienced with translating knowledge into constraints. This complexity issue has led to the creation of authoring tools designed to assist with ITS development. In 1997 A. Munro et al. at the University of Southern California developed RIDES, a system that enables users to easily author simulation-based graphical tutors (Munro et al., 1997). RIDES provides support for domain modeling, contains editors for creating graphical components, and generates rules that drive the simulation. RIDES was successfully used to create tutors for medical devices, shipboard radar and other equipment. The primary limitation of RIDES is that it only supports a causal domain model without support for conceptual knowledge, which means it is only suitable for modeling physical devices and their operation.

REDEEM is another landmark ITS authoring tool developed by N. Major et al. at Nottingham University (Major and Ainsworth, 1997). While prior ITS authoring tools significantly accelerated the creation of ITSs, the authoring tools themselves still required significant expertise. REDEEM was designed to integrate concepts from psychology with ITS development to provide an authoring environment that allows teachers with limited ITS experience to create effective tutors. REDEEM was successfully used to create a number of ITSs, including a genetics tutor for 15-year-old high school students. One major limitation of REDEEM is that while it does maintain a student model at a high level to dynamically select learning activities based on the student's performance, it does not customize the learning activities themselves once they have been selected.

The ICTG at the University of Canterbury have also developed authoring systems for their newer constraint-based tutoring approach. WETAS is an authoring tool aimed at experienced ITS developers, while ASPIRE is a newer web-based system that caters to developers and teachers alike (Mitrovic et al., 2008). Using ASPIRE, the ITS author first models the domain as an ontology of related concepts, then describes the structure of tasks to be completed by the students, and finally provides a set of sample problems and solutions. The authoring system then examines the data provided by the teacher and uses a machine learning algorithm to automatically generate the constraints. Once the tutor has been designed in the authoring interface, it can be easily deployed to a separate tutoring server, which the students are able to access. ASPIRE supports a range of graphical and text-based interfaces, and can use its student model to adapt the tutoring experience uniquely to each student. After testing ASPIRE with a number of teachers, The ICTG researchers have found that their tool dramatically reduces the development time of a constraint-based ITS, and that it is capable of producing robust tutoring systems.

2.3 Intelligent Augmented Reality

While there have been a number of studies exploring the combination of intelligent tutors with virtual reality (Mendez and Herrero, 2004; Evers and Nijholt, 2000), there has been minimal prior research exploring the combination of Intelligent Tutoring Systems with Augmented Reality. A few studies claim to have created AR applications that are intelligent, but in practice these systems are minimally intelligent and do not employ domain, student and pedagogical models to provide a unique tutoring experience for each user. For example, Y. Qiao et al. developed what they call an AR Intelligent Tutoring System that teaches users about the instruments and dials in a cockpit (Qiao et al., 2008). Their definition of intelligence in this context stems from the fact that their system detects which cockpit component the user is looking at and then displays relevant information describing the component's function. This context-based display interface is very different from the kind of intelligence that is employed in the robust ITSs described in section 2.2.

S. Feiner et al. developed a prototype in the 1990s that employed what they call Knowledge-based Augmented Reality (Feiner et al., 1993). Their system employed a rule-based intelligent back-end called IBIS, which stands for Intent-Based Illustration System. They used IBIS to dynamically generate graphics based on the communicative intent of the AR system at any particular moment. The communicative intent is represented by a series of goals, which specify what the resulting graphical output is supposed to accomplish. For example, a goal could be to show a property of an object, such as its location or shape, or to show a change in a property. Feiner and his colleagues demonstrated their system with a prototype that assists users with laser printer maintenance. While this system is intelligent in how it generates the graphics that are displayed to the user, it is not intelligent from a training or tutoring standpoint. The system does not model student knowledge, and thus the experience is the same for each user.

Chapter 3

Research Approach

The primary focus of this masters thesis project is to explore the combination of Augmented Reality with Intelligent Tutoring Systems. AR has proved to be an effective medium for visually conveying abstract knowledge and spatial concepts—particularly when the virtual content is combined with the real environment in a meaningful way. Assembly and maintenance tasks are an excellent application domain because the virtual content augments the user's view of reality in order to assist with a real world task. Both the real and virtual content is meaningful, and they are shown together in context.

While there has been much research into the use of AR to assist with assembly and maintenance, existing systems generally focus on improving user performance while using the AR interface as opposed to teaching the user how to perform the task without assistance. Most systems guide the user through a fixed series of steps and provide minimal feedback when the user makes a mistake, which is not conducive to learning.

Intelligent Tutoring Systems are a compelling means of applying Artificial Intelligence in education. They generate a customized experience that is tailored to the strengths and weaknesses of each user, and they provide highly detailed feedback that helps students learn from their mistakes and master the subject matter. ITSs have been created for a wide variety of domains, but the interfaces employed are normally text-based or simple 2D graphical applets, which limit their ability to convey spatial or physical concepts. The integration of Augmented Reality interfaces with Intelligent Tutoring back-ends creates new possibilities for both fields and could improve the way we acquire practical skills and associated knowledge.

3.1 Research Questions

There are a number of research questions that provide the motivation for this masters thesis. This section describes some of the fundamental questions and how they are addressed.

How and to what extent can Augmented Reality-based training benefit from the use of intelligent tutoring approaches?

This is the overarching question that drives my research. My hypothesis is that AR training can benefit significantly from ITS approaches, but I must support my claim with valid reasoning and evaluation. At each stage of the project, I considered how to best integrate Intelligent Tutoring approaches into my AR training prototype. It is likely that some aspects of Intelligent Tutoring Systems are more applicable in the AR domain than others. In order to maximize the performance of the prototype, I selected an effective combination of techniques from both domains.

What type of Intelligent Tutoring System is best suited for use with Augmented Reality-assisted assembly and maintenance tasks?

As part of my initial research, I compared multiple Intelligent Tutoring approaches in order to decide which type is best suited to my project. Tutoring systems based on production rules lend themselves more naturally to procedural training tasks, while constraint-based systems are generally more powerful when it comes to modeling and applying domain knowledge. Choosing the right tool for the job is important, and I investigated several different ITS solutions to find the best fit for my prototype.

Can intelligent Augmented Reality-based training enable users to learn and retain assembly and maintenance skills more effectively when compared with traditional AR training approaches?

This question has not been well-addressed by prior research. Previous studies have observed improved user performance while using an AR interface when compared with other training methods, but they have not attempted to address how well the users learn or retain knowledge after the training is complete. The Intelligent Tutoring back-end of my prototype allows the system to enhance learning effectiveness and improve skill retention over non-intelligent AR approaches. I attempt to measure this difference in my evaluation by switching between the ITS back-end and a more traditional approach that follows a rigid procedure.

3.2 Prototype

In order to address the research questions outlined in section 3.1, I developed a software framework that combines Augmented Reality-assisted training with Intelligent Tutoring approaches. The system is designed to be as modular as possible so that it can be easily adapted for new assembly and maintenance tasks. The display elements and ITS domain model must be customized for each new scenario, but the underlying software architecture, scaffolding algorithms and other back-end processing remains the same.

The primary task used for evaluation involves training users to assemble components on a computer motherboard. This includes sub-tasks like identifying individual components, installing memory, processors and heat sinks. An AR motherboard assembly training prototype was created by K. Baird and W. Barfield, but their system did not employ spatially-registered AR and did not utilize Artificial Intelligence when it prompted users to follow a rigid series of assembly steps (Baird and Barfield, 1999). In addition, their evaluation concerned itself only with the performance of users while using the display and did not test knowledge retention after the training was complete.

Why Computer Motherboard Assembly?

There are a variety of reasons why intelligent AR-assisted computer motherboard assembly is an ideal prototype system for my masters thesis. From a technical standpoint, motherboard assembly lends itself well to the use of Augmented Reality. The majority of computer vision tracking algorithms involve estimating the pose of flat surfaces. Motherboards and their components, such as memory, processors and graphics cards are generally planar in nature with straight edges and distinct features that make them easy for a computer to recognize and track. The task itself has a good level of complexity—it is more realistic and practical than assembling toy blocks, yet it is not overly complex within the scope of a masters thesis. The general motherboard assembly task can easily be divided into a number of procedural subtasks, which makes it scalable and modular in nature.

Resource availability is a major consideration when conducting any kind of research. During the project period, I was based at the Human Interface Technology Lab New Zealand (HITLabNZ) within the University of Canterbury. The hardware and software required for the prototype (motherboards, head-mounted displays, tracking software, etc.) is readily available at the HITLabNZ, and the lab is has a wealth of expertise in the area of Augmented Reality. Similarly, the Intelligent Computer Tutoring Group (ICTG) is also based at the University of Canterbury, where researchers have been studying Intelligent Tutoring Systems for more than 15 years. Another resource to consider is the availability of test subjects to evaluate the prototype. At the university I had access to a wide variety of participants from different backgrounds. Computer motherboard assembly is unfamiliar to many people, yet it is not so specialized that few would want to learn about it.

Chapter 4

ITS Design

As previously mentioned in section 2.2, Intelligent Tutoring Systems have been developed for a wide range of topics, including physics, genetics, programming languages and English grammar. Generally speaking, the interfaces used with ITSs have been centered around text, web forms, or simple 2D graphical widgets in the style of Java applets. Thus, there are a variety of new design issues to be considered when developing an Intelligent Tutoring System to be used with a 3D augmented reality interface. This chapter covers the ITS design process—beginning with identifying desirable ITS features, evaluating existing ITS authoring solutions with respect to these characteristics, and justifying ASPIRE (Mitrovic et al., 2008) as the final choice. Section 4.3 discusses the ITS creation in more detail and describes the process of modeling the motherboard assembly tasks in ASPIRE.

4.1 Desirable ITS Characteristics

This section describes various ITS properties that are desirable for an AR training task. Some of the following items pertain to more than just AR interfaces, and are good features to have in any ITS.

4.1.1 Flexible Data Representation and Communication

To enable an AR interface to be coupled with an ITS back-end, the ITS must be flexible in the way that it represents and communicates information. Many ITS solutions are designed to interpret input from 2D interface components such as windows, text fields, buttons and sliders. The input itself typically consists of simple data types such as strings, integers and floating point numbers. An AR interface deals with higher-level data such as spatial position and orientation, movement, 3D graphics and gesture recognition. Of course, all of this is ultimately represented as a collection of lower-level data types, but an ideal ITS would be able to directly interpret higher-level structures such as 3D transformation matrices, physical/material properties and various methods of interaction beyond the traditional mouse and keyboard.

More important than the data representation is the fact that the ITS must be able to send and receive information from the AR interface. Ideally the ITS would act as a server that can communicate over a network—it receives cues from the AR interface, makes decisions based on the cues and sends the appropriate feedback to be displayed in the AR view. This has the added benefit of allowing the ITS and AR interface to run on separate machines. Many ITS solutions do not offer this degree of flexibility with regard to communication, and are thereby locked in to a particular style of desktop or web-based 2D graphical interface. For example, the ITS might be hardwired to listen for events related to mouse clicks and keyboard input, and therefore would not be well-suited to a 3D AR task.

4.1.2 Learner Control

There is a significant amount of research suggesting that students benefit most when there are fewer restrictions imposed on the tutoring process. Providing students with the freedom to explore and arrive at correct solutions via their own cognitive processes can improve the learning outcome (Gilbert et al., 2009). This goal can be addressed by support for non-procedural tasks, flexible solution representations, and scalable feedback, which are discussed in detail below.

4.1.2.1 Support for Procedural/Non-Procedural Tasks

Procedural tasks have a defined algorithm or method for completion. For example, there is a procedure to be followed when assembling furniture, adding fractions or carrying out a chemical reaction. There may be some flexibility in the ordering of individual steps, but there is still a defined process that arrives at a valid solution. Non-procedural tasks are less clearly defined and involve more freedom of thought. For example, writing an essay or designing a database are more open-ended tasks that do not have an exact algorithm or series of steps to be followed. There may be guidelines, such as including an introduction at the beginning of an essay and including a topic sentence at the beginning of each paragraph, but these constraints are still quite general and do not describe an exact process.

AR-based training for assembly and maintenance traditionally involves more procedural tasks that instruct the student to follow a series of defined steps. However, one can imagine an AR tutor that teaches students to plant a garden or perform a similar less procedural task. The tutor may give general guidelines, such as planting certain species next to each other or how deep to dig, but it would not give the student a fixed series of steps describing exactly where each plant should go. For this reason, the chosen ITS would ideally be able to support both procedural and non-procedural tasks, even though the majority of AR assembly tasks are procedural.

4.1.2.2 Accepting Multiple Solutions

Another important property to consider when evaluating an ITS is its degree of flexibility with regard to the representation of solutions. As previously mentioned in section 2.2.1, traditional rule-based (model-tracing) systems represent solutions as a collection of problem-solving procedures. They do support multiple correct solutions for a single task, but each solution must be anticipated in advance by the ITS developer and have an associated set of rules describing the procedure to be followed to arrive at the solution. Constraint-based tutors, on the other hand, describe abstract features of correct solutions via constraints. They do not detail the exact procedure for satisfying the constraints, and thus are more flexible in terms of accommodating multiple solution paths. The ITS developer does not need to anticipate all correct solution procedures in advance.

While some AR-based training tasks follow a set of defined procedures, they could also potentially benefit from the use of the more flexible solution representations found in constraint-based tutors. For example, when assembling a table, it doesn't necessarily make sense to follow a strict procedure to attach each of the four legs. It would be possible to create a set of rules for each possible solution procedure, but that would be tedious and there might be valid solution paths that are overlooked.

In contrast, a constraint-based approach would simply state that the table must have four legs attached in the proper locations and orientations. It would then be left up to the AR interface and the student to devise a method to meet the solution requirements set forth by the tutor. If the ITS developer wanted the student to follow an exact procedure, this can still be achieved by adding additional constraints that detail a more strict solution path.

4.1.2.3 Flexible Teaching Strategy

Scaffolding is an educational concept that entails scaling the level of assistance with a task as needed based on how a student performs (Wood et al., 1976). When a student is learning a concept or skill for the first time, the system provides detailed assistance with each aspect of the learning task. As the student becomes more familiar with the task, the level of assistance is gradually scaled back until the student is able to perform the task without assistance. If the student forgets something or makes a mistake, the scaffolding kicks in and reminds the student of the proper procedure. The goal is to reduce reliance on the system over the course of the training period so that eventually the student can complete the task without assistance.

Effective ITSs are able to adjust their teaching strategy to suit each student. Prior research suggests that novice students benefit most from immediate intervention and feedback when mistakes are made, while more knowledgeable students benefit from less rigid instruction, which allows them the opportunity to discover and correct their own mistakes (Gilbert et al., 2009). In general, students should be given the minimum level of assistance required to successfully complete the task (via scaffolding), and a minimal level of feedback in the form of hints when a mistake is made. It is important for any ITS to be able to accommodate different styles of instruction depending on the task to be performed and the student's level of expertise.

For example, in the case of AR assembly, the tutor might begin by asking the student to complete the task step without any visual cues, and if it determines that the student is having trouble it would then intervene and guide the student to perform the correct procedure. In terms of feedback, the ITS could intervene as soon as it detects that the student is about to make a mistake, or it could allow the student to make the mistake and then describe what went wrong. The ITS would ideally be able to accommodate multiple teaching strategies like these in order to best teach a particular task to a particular student.

4.1.2.4 Adaptive Student Model

One key characteristic of the most effective intelligent tutoring systems is that they not only maintain a model of the student's cognition as tasks are completed, but they actively use the student model to dynamically influence the teaching strategy on the fly. A student model typically contains information pertaining to the student's knowledge and performance on various tasks. Many ITSs merely use this data to supply a report at the end of the tutoring session that describes the student's performance on various sub-tasks. More full-featured ITS solutions use the student model to adapt instruction on the fly to the student's knowledge level and learning style. For example, if the student struggles on a particular sub-task, the ITS may provide more assistance and revisit that task again in the future. If the student completes the task with no assistance and without difficulty, the ITS might consider the student to be an expert in that particular area and could focus on other areas with which the student is less familiar. An adaptive student model that influences instruction is a desirable feature for any ITS regardless of the subject being taught or the interface used.

4.2 Survey of Existing ITS Authoring Solutions

As previously mentioned in section 2.2.3, Intelligent Tutoring Systems can be incredibly complex and time-consuming to create. Depending on the type of tutor, it can take 300-1000 hours of development time to produce one hour of learning material (Murray, 1999). Thus, for the purposes of the master's thesis project, an existing authoring system was chosen in order to to rapidly produce the intelligent tutoring system for the AR training task. In section 4.1 I outlined a number of desirable ITS features for an AR assembly task. In this section I evaluate several existing authoring systems with respect to these criteria and justify my final choice.

4.2.1 CLIPS

In my survey of authoring systems, I examined a number of solutions that produce different classes of tutors—some more intelligent than others. Some of the more primitive tutors are based on expert systems, which are a predecessor to ITSs. Expert systems consist of a model of expert knowledge that is imparted upon the student by guiding the student through a rule-based procedure. These systems are meant to solve a given problem, rather than teach, and thus do not concern themselves with whether learning is actually taking place. More specifically, they usually do not maintain a student model to track progress and adjust the teaching strategy accordingly.

CLIPS is a software library that handles the development and execution of expert systems (Wygant, 1989). The primary advantage of using CLIPS is that it is very flexible and can be easily integrated with any sort of interface, including Augmented Reality. Its rule-based approach lends itself well to procedural tasks such as AR-assisted assembly, although it would not be well suited to non-procedural AR tasks. To make up for the lack of intelligence in expert system design, I could use CLIPS to handle the basic tutoring and could separately implement my own student and pedagogical models to make it more adaptive. However, I encountered other authoring systems that produce more intelligent tutors with student models and pedagogical controls built in. Another disadvantage of CLIPS is that the expert model and set of rules must be created manually, as opposed to other authoring systems that can automatically generate rules or constraints based on examples provided by the author. Furthermore, due to the strict procedural rule-based approach, the author must anticipate all solution paths in advance, and thus the system is incapable of accepting unanticipated correct solutions.

4.2.2 RIDES

RIDES is a tutor authoring system developed by A. Munro et al. at the University of Southern California (Munro et al., 1997). It produces simulation-based graphical tutors that enable students to learn about physical devices and their operation. The developer can use the authoring interface to outline the domain model, design graphical components and create rules that drive the simulation. The authoring system can observe the author's behavior with the simulated model and "learn by example", thereby automatically inferring the rules that govern the interaction. RIDES has been successfully used to create tutors for medical devices, shipboard radars and other complex equipment.

While the physical simulation-based approach of RIDES could be useful for an AR assembly task, it is not a good solution for this project. One fundamental issue is that the tutoring back-end cannot be easily decoupled from the graphical front-end. This means that integrating RIDES with an AR interface would be a difficult task since it is designed to work with the integrated 2D graphical interface. Another problem with RIDES is that it only supports a causal domain model, without support for more abstract conceptual knowledge, which means that it is limited to modeling physical devices and their operation. While an AR assembly task is primarily a physical endeavor, there is still a significant amount of conceptual knowledge that could be conveyed in order to provide context for the student's actions.

4.2.3 REDEEM

REDEEM was developed by N. Major et al. of Nottingham University (Major and Ainsworth, 1997). It was created with the goal of integrating concepts from psychology with ITS development in order to make a truly accessible ITS authoring system. While prior authoring systems accelerated the ITS development process, they were still not realistically usable by teachers with limited ITS experience. REDEEM primarily allows teachers to automate and dynamically control the presentation of existing learning material. The author divides the learning material into individual pages that are presented to the student based on the selected teaching strategy. The tutor receives input from the student in the form of answers to multiple-choice questions, and can then use the student's answers to dynamically alter the teaching strategy.

While REDEEM has been successfully used to produce tutors on various topics, including a genetics tutor for high school students, it is not well suited to AR assembly tasks for several reasons. Firstly, like RIDES, it was not designed to work with custom interfaces, and it would be difficult to decouple the interface built into REDEEM from the tutoring back-end. Secondly, the fact that REDEEM can only react to input in the form of multiple choice answers severely limits the interactivity that it could provide for an AR assembly task. Clearly this also restricts its ability to interpret unanticipated solutions or solution paths. Finally, REDEEM doesn't truly support the modeling of domain knowledge—it merely presents existing material based on a set of properties configured by the author. This means that the tutor is incapable of understanding the concepts and interrelationships in the material it presents, which limits the intelligence of its tutoring approach.

4.2.4 xPST

The Extensible Problem-Solving Tutor (xPST) was a collaborative effort between S. Gilbert et al. of Iowa State University and the University of Tampa (Gilbert et al., 2009). Unlike some of the other authoring solutions presented in this section, xPST was designed specifically with the intent of providing a modular ITS back-end that can be coupled with an existing front-end graphical interface with relative ease. xPST produces model-tracing tutors, where the cognitive model describes objects within the learning domain and rules are used to govern the interaction of the objects. In general, every relevant interface element is mapped to an object, and has one or more rules associated with it. The curriculum module contains a set of tasks to complete within the interface. xPST employs an event manager module to eavesdrop on user actions and send them to the tutoring engine, which then supplies feedback to be displayed in the interface. The tutor is capable of running as a server application, which communicates with the interface over TCP/IP.

xPST is a reasonable authoring candidate for an AR assembly tutor, but it does have some limitations. One such drawback is the fact that it does not appear to have significant support for student modeling. The tutor generally carries out the instruction in the same way regardless of how the student is performing. It does provide feedback when a mistake is made, but it does not use this information to alter the teaching approach. In addition, xPST does not support automatic rule generation—the author must create the rules manually, which as previously mentioned can be an incredibly time-consuming endeavor. Finally, like other model-tracing tutors, xPST relies on production rules, which means that it is incapable of accepting unanticipated solutions.

4.2.5 CTAT

The Cognitive Tutor Authoring Tools (CTAT) is a suite of utilities created by V. Aleven et al. of Carnegie Mellon University (Aleven et al., 2006). CTAT supports the creation of both "cognitive tutors", which employ a domain model, and "example-tracing" or "pseudo" tutors. Example-tracing tutors can be created without programming or manual rule creation—the author demonstrates example solutions, generalizes them, and then annotates them with hints and feedback to be provided to the student. The CTAT suite also contains modules for evaluating the educational impact of individual tutor features by automatically creating comparison tests between a tutor with the feature included and an otherwise identical tutor with the feature omitted. CTAT supports the use of alternative interfaces, and has been successfully used to instruct students on the use of existing applications such as Macromedia Flash 2004.

While CTAT is considerably more flexible than other ITS authoring solutions such as RIDES and REDEEM, it still has some limitations that make it a poor candidate for use with an AR assembly interface. While CTAT does support some custom interfaces, it is currently limited to communicating with a few GUI tools such as Java Netbeans and Flash. The tutoring engine is primarily designed to interpret 2D interface interaction such as button clicks and keyboard presses, so it would be difficult to adapt CTAT to a 3D AR environment. The automated rule generation and solution generalization features are desirable, but they are inconsequential if the tutor cannot communicate with the AR interface.

4.2.6 XTA

The eXtensible Tutor Architecture (XTA) was developed by G. Nuzzo-Jones et al. of the Worcester Polytechnic Institute (Nuzzo-Jones et al., 2005). It is an authoring platform for creating and deploying model-tracing and example-tracing ITSs across many different platforms. Like xPST and CTAT, XTA strives to separate the ITS logic from the chosen presentation interface. The ITS logic is divided into the Curriculum Unit, which describes the set of material to be taught, the Problem Unit, which represents an individual problem to be tutored, and the Strategy Unit, which allows for high-level control over the problems flow and teaching approach. XTA does provide a form of adaptive instruction, where submitting an incorrect solution can result in covering additional material on that particular concept to reinforce the correct behavior.

While XTA is a very flexible ITS authoring tool, it suffers from the same limitations as CTAT. The list of supported interfaces currently consists of Java Swing, WebStart and HTML. These are tools for creating 2D interfaces and thus they cannot be used for an AR application. Its adaptive instruction features make XTA superior to CTAT, but again this is inconsequential if the tutor cannot communicate with an AR interface.

4.2.7 ASPIRE

ASPIRE was developed by A. Mitrovic et al. of the Intelligent Computer Tutoring Group at the University of Canterbury (Mitrovic et al., 2008). ASPIRE produces constraint-based tutors, which use constraints instead of production rules to evaluate student solutions. One significant advantage of the constraint-based approach is that ASPIRE is able to accept solution paths that were not anticipated in advance by the ITS author. More information regarding constraint-based tutors can be found in section 2.2.2.

The authoring interface in ASPIRE is web-based, and the ITS creation process consists of a series of defined steps. The ITS author first models the problem domain as an ontology of related concepts, then describes the structure of tasks to be completed by students, and finally provides a set of sample problems and solutions. The authoring system then employs a machine learning algorithm to generalize the sample problems and solutions and automatically generates constraints. Once the tutor has been designed in the authoring interface, it can be deployed to a separate tutoring server, which the students can access.

Among the tutor authoring solutions listed here, ASPIRE is the optimal choice. It has virtually all of the desired ITS characteristics outlined in section 4.1. ASPIRE supports both text-based and graphical interfaces, and also supports communication over a network via a remote procedure call (RPC) protocol, which would allow it to communicate with an external AR interface. In addition, it has full support for student modeling and multiple teaching strategies, as well as customizable levels of feedback. ASPIRE can also handle both procedural and non-procedural tasks, which is another factor that distinguishes it from the other ITS authoring solutions. As previously mentioned, the constraint-based approach allows ASPIRE to handle unanticipated solution paths, and it automates the generation of constraints, which significantly reduces the ITS development time. One drawback of ASPIRE is that it is not designed to directly interpret spatial or graphical information such as 3D transformation matrices or material properties, which would be useful for an AR tutoring system. However, ASPIRE does support many low-level data types such as integers, floats and strings, which can be combined to represent more complex entities.

4.2.8 Survey Summary

Table 4.1 on the next page provides a summary of the survey results. ASPIRE is the only system that meets all of the criteria outlined in section 4.1.

Table 4.1: Comparison of ITS Authoring Solutions

Authoring System	Authoring Output	Procedural Tasks	Non-Procedural Tasks	Supports External Interface	Adaptive Instruction	Accepts Unanticipated Solutions	Automated Knowledge Generation
ASPIRE	Constraint-Based ITS	Yes	Yes	Yes	Yes	Yes	Yes
RIDES	Simulation of Physical Devices	Yes	No	No	Yes	No	Yes
REDEEM	Teaching Strategies	Yes	No	No	Yes	No	No
CLIPS	Rule-Based Expert System	Yes	No	Yes	No	No	No
xPST	Model-Tracing Tutor	Yes	No	Yes	No	No	No
CTAT	Example-Tracing Tutor	Yes	No	No	No	No	Yes
XTA	Model-Tracing Tutor	Yes	No	No	Yes	No	Yes

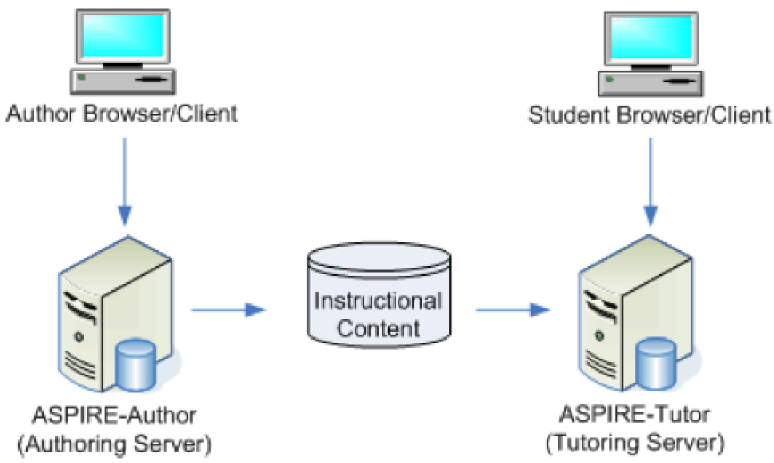


Figure 4.1: ASPIRE Architecture (Mitrovic et al., 2008).

4.3 Modeling the Assembly Task in ASPIRE

The ASPIRE system consists of two primary components—ASPIRE-Author and ASPIRE-Tutor. ASPIRE-Author is a server that provides a web-based interface for creating constraint-based tutors. After a tutor is created, it is deployed to the ASPIRE-Tutor server, which is accessed by students. Figure 4.1 conveys the high-level architecture of the ASPIRE system (Mitrovic et al., 2008). Modeling the motherboard assembly task in ASPIRE-Author consisted of several steps, which are described over the following subsections.

4.3.1 Domain Model

The first stage of the authoring process involves describing characteristics of the teaching domain and composing an ontology of related concepts within the domain. Specifying the domain characteristics includes indicating whether the domain is procedural or non-procedural. If it is procedural, the author is required to enumerate the general problem-solving steps associated with completing tasks in the domain. In the case of the motherboard assembly tutor, the assembly task is procedural in nature and has a discrete set of steps to be completed such as opening the processor enclosure and inserting the processor in the correct orientation. The system still allows for the flexibility of multiple solutions, but in this case the solutions all conform to a procedural structure.

After specifying the domain characteristics, the designer uses a java applet em-

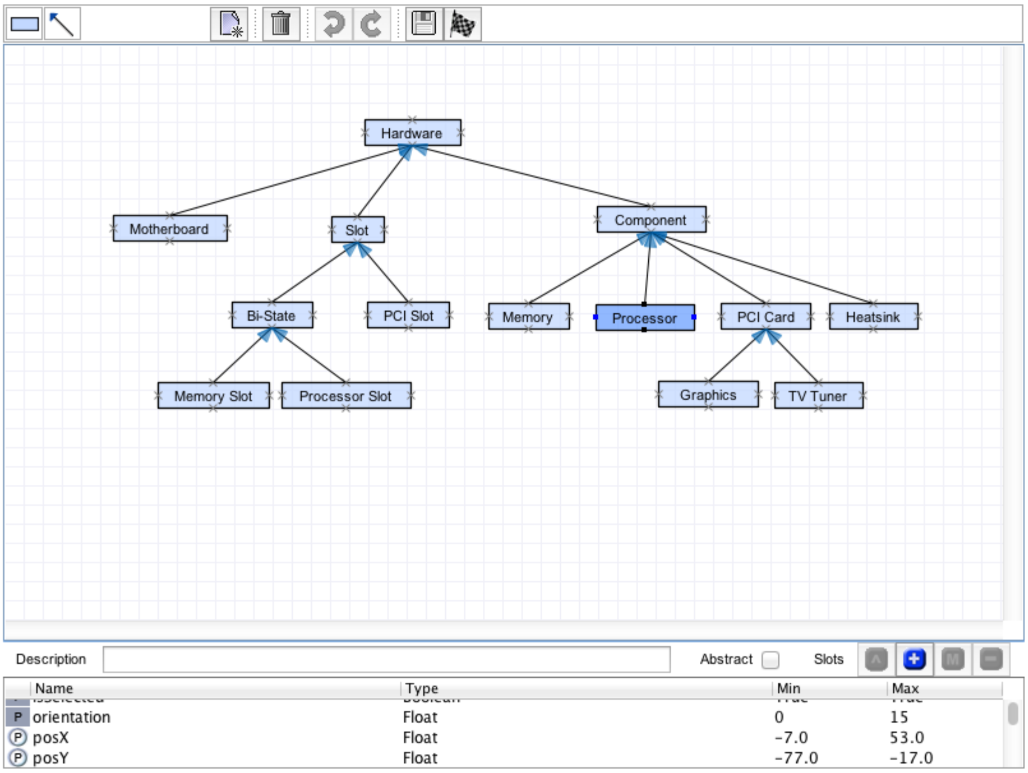


Figure 4.2: Composing the domain ontology in ASPIRE.

bedded in the web-based authoring interface to model important concepts in the domain, their properties, and the relationships between the concepts. A knowledge domain often encompasses a wide range of concepts, and in this step the author must determine which concepts are directly related to the problems the students will solve using the tutor, eliminating those which are not necessary. Figure 4.2 shows the domain ontology that was created for the motherboard assembly tutor. The author creates an entity for each concept and then specifies properties associated with the concept. For example, in the case of a memory slot on the motherboard, an important property to include is an indicator of whether the slot is open or not, since the slot must be opened before the memory can be installed. This property can be represented as a true/false boolean value, and the tutor utilizes this information when teaching the student to install the memory. After creating an entity, the author attaches it to the hierarchical structure of all of the domain concepts by connecting it to another concept with a directional link that indicates an "is-a" relationship. Properties of parent concepts are inherited by their children.

4.3.2 Problem/Solution Design

After composing the domain ontology, the author specifies the structure of the problems and solutions by indicating which ontology concepts are involved with each problem-solving step. For example, installing computer memory involves four steps: Identifying and picking up the memory component, opening the locking levers at the ends of the memory slot on the motherboard, aligning the memory with the slot in the correct orientation, and pushing the memory down into the slot until it locks. Each of these steps has at least one domain ontology concept associated with it, and each concept has properties that are used to determine whether the student's solution for each step is correct. In the case of the "open locking levers" step, the ITS uses the boolean "isOpen" property of the "Memory Slot" concept to determine whether the slot has been successfully opened or not. The value of the boolean property is set via the AR interface, which is described in the next chapter.

After designing the problem/solution structure, the next step of the design process prompts the author to create the interface that the students will see when they login to the tutor. Most tutors created with ASPIRE have web-based interfaces, and this stage allows the author to create the layout of graphical components that appear on the tutoring web page. In the case of the AR motherboard assembly tutor, the students do not use the built-in web-based interface. Instead, the AR front-end communicates with ASPIRE directly over a network.

4.3.3 Creating Problems and Example Solutions

After designing the problem/solution structure, the author uses the authoring interface to create at least one problem to be presented to students, along with one or more example solutions for each problem. Each problem fits the structure specified in the previous authoring step, but has different values in the problem statement. For example, a fraction addition tutor might have the following problem-solving steps: Find the lowest common denominator, add the numerators together, simplify the resulting fraction if necessary. Every fraction addition problem follows this structure, but each problem and solution deals with different values, e.g. "Add

1/2 and 2/3". In the case of the motherboard assembly tutor, the problem structure describes steps that apply to all motherboards, while a particular problem and associated solutions apply to a specific brand and model of motherboard.

A key feature of ASPIRE is that it allows multiple solutions to be specified for each problem. In the case of motherboard assembly, there is often only one way to correctly install each component, but this is not always the case. For example, a memory module can be inserted into one of several slots, and a heat sink with a symmetrical design can sometimes be installed in more than one orientation. Accepting these different configurations as correct solutions gives the student more flexibility when solving the problem, which can enhance the learning outcome.

4.3.4 Constraint Generation

After all of the problems and solutions have been specified, the authoring system automatically generates the domain model, which consists of all of the constraints that describe the problems and solutions within the teaching domain. The constraint generation process utilizes a machine learning algorithm that generalizes and specializes constraints as necessary based on the solutions provided by the author. If only one valid solution has been provided for a problem, the constraints will be very specific to that solution, while supplying multiple correct solutions will result in more general constraints.

ASPIRE utilizes two types of constraints. Syntax constraints check whether the student's solution is in the correct form and that it follows the syntactic rules of the domain; they ensure that all of the necessary solution components are specified, that the correct data types are used, and that the components are related to other solution components as necessary. Semantic constraints, on the other hand, are used to check the validity of the solution once the syntax requirements have been met (Mitrovic et al., 2008). For example, a syntax constraint might specify that a floating point value representing the memory module's orientation must be included in the solution for the memory insertion step, while a semantic constraint would check to make sure that the supplied orientation value is acceptable and constitutes a valid solution. If an integer value is supplied for the orientation instead of a floating point, or if no value is supplied at all, the syntax constraint is violated and an appropriate feedback message is relayed to the user interface. If the orientation value is supplied correctly, but the value itself is incorrect, the semantic constraint is violated and a separate feedback message is provided.

After the constraints are generated, the author can use the authoring interface to edit the constraints by hand, as well as change the default feedback messages to something more user-friendly. Rather than saying "The y component of the position of the memory should be greater than 25.0", this default message could be changed to say "The memory is not positioned correctly. The correct location is further to the right", which is clearly more intuitive to a human student.

4.3.5 Tutor Deployment

When the domain model has been generated and the author is satisfied with the constraints, the tutor is ready to be deployed to ASPIRE-Tutor, where it can be accessed by students. The author can create individual accounts for students and add them to groups, each of which can have customized settings. These settings include specifying the type of feedback to be supplied as well as the progression between the feedback levels as the student makes mistakes. Typically the system is set to provide minimal help initially by only indicating that that student's solution is incorrect, but not showing how it is wrong. The tutor then provides more and more detailed feedback hints as the student struggles with the problem until finally the full solution is revealed. While this is the default behavior, the author can alter this structure to provide more or less challenge to different groups of students as necessary.

4.3.6 ASPIRE Summary

The ASPIRE authoring process allows robust constraint-based ITSs to be created rapidly and without the need for programming. Rather than engaging in the time-consuming process of creating individual rules or constraints, the author simply outlines the important concepts in the knowledge domain, creates the problem and solution structure, and provides examples of correct solutions. From these examples and the domain ontology, the system automatically generates the constraints that govern the tutoring process. A total of 275 syntactical and semantic constraints were generated for the motherboard assembly tutor, which potentially represents hundreds of hours saved from manual programming and testing.

In addition, the ASPIRE system automatically handles student modeling. The tutor keeps track of each student's performance, even between multiple tutoring sessions, and can customize the tutoring experience for each person. This includes providing varying levels of feedback as well as selecting problems based on the user's current level of knowledge or skill. Teachers are provided with statistics about their students, including details of which constraints are being violated as well as automated generation of learning curves that show how each student performs over time. ASPIRE is also capable of communicating with an external interface over a network connection. This gives it the flexibility to be integrated into a variety of tutoring environments, including Augmented Reality. The next chapter discusses the design of the AR interface that connects the ITS with the student.

Chapter 5

AR Interface Design

This chapter covers the design of the Augmented Reality tutoring interface, which relays information between the Intelligent Tutoring System and the student. The flow of information moves in both directions. The interface visualizes instructions from the ITS in the form of 3D graphics, animations, audio and text, which are seamlessly blended with the student's view of reality via a head-mounted display. The interface then uses a camera to observe the student's behavior and relays information back to the ITS. The ITS analyzes the data, provides feedback about the student's performance and makes decisions about what material to present next. In this way, the ITS acts as the brain of the system, while the AR interface serves as the mouth and sensory organs that allow the brain to perceive and communicate with the world.

This chapter begins by discussing the software architecture and the visual elements that are displayed to the user. The first section also covers how the Augmented Reality interface communicates with the Intelligent Tutoring System over a network connection via remote procedure calls. The chapter concludes by describing the hardware setup, including the head-mounted display and the computer motherboard that was used for the prototype assembly task.

5.1 Software Architecture

The primary goal when designing the software architecture was to create a flexible system that can be applied to a wide range of assembly and maintenance

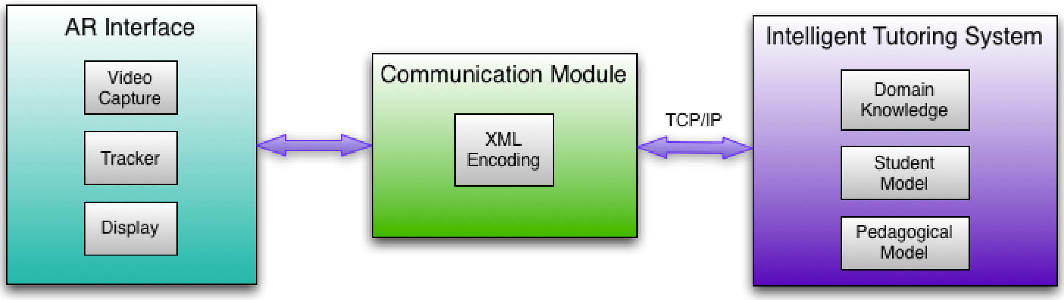


Figure 5.1: High-level software architecture.

tasks. The resulting software library is application-independent, and this general framework was then used to create the interface for the computer motherboard assembly prototype. All of the code that is specific to the motherboard assembly tutor resides in a separate location from the framework, and this code can be changed to accommodate a different training scenario with relative ease.

Another major design goal was to divide the framework into distinct modules that work independently and communicate with each other as needed. This allows the implementation details of each module to be abstracted from the others, allowing changes to be made with minimal impact on other parts of the system. For example, the underlying tracking algorithm could be changed from a marker-based approach to a natural feature-based approach without affecting the display or communication modules. Similarly, a different computer graphics library could be substituted in the display module without affecting the others. This modular design minimizes the effort required to maintain and extend functionality by allowing code to be reused whenever possible.

Figure 5.1 illustrates the high-level layout of the software architecture. At the back-end, The Intelligent Tutoring System encapsulates the domain knowledge, student model and pedagogical model, which allow it to control the teaching process and provide a customized experience for each student. For further details regarding the design and implementation of the ITS, please refer to Chapter 4. The Augmented Reality interface, at the front-end, encapsulates the video capture, tracking system, display and keyboard input, which are covered in detail later in this chapter. The communication module serves to relay messages between the AR interface and the ITS via XML remote procedure calls over a TCP/IP network connection. This allows the front-end and back-end modules to reside on separate machines for increased flexibility.

5.1.1 Pre-existing Software Libraries

Creating an Augmented Reality interface completely from scratch would be far beyond the scope of a masters thesis project. As such, a number of pre-existing software libraries were used to streamline the development process. This section provides a brief outline of all major third-party libraries used to create the interface. The libraries are discussed in further detail later in the chapter.

- **osgART**[1] (Looser et al., 2006) - Framework for creating Augmented Reality applications in C++ that combines computer graphics, video capture and tracking into a single package.
- **XML-RPC For C++ on Windows**[2] - Lightweight software library that enables client code to execute functions on a server using XML remote procedure calls over HTTP.
- **XMLParser**[3] - A simple C++ class that converts XML strings to a hierarchy of nodes (and vice versa) for easy access to properties and values.
- **Microsoft Speech API**[4] - Package included with Windows that provides text-to-speech support for programs written in C.

5.2 Tracking Module

The tracking module enables the system to perceive the position and orientation of the computer motherboard and its components relative to the camera affixed to the head-mounted display. This serves two fundamental purposes: Firstly, it allows the display module to render 3D graphics on top of the video frame in such a way that the virtual models appear to reside in the real world. Secondly, the tracking module relays information about the relative positions of the motherboard

[1] osgART - www.osgart.org, retrieved 7/02/2012
[2] XML-RPC - www.xmlrpc-c.sourceforge.net/windows.php, retrieved 7/02/2012
[3] XMLParser - www.applied-mathematics.net/tools/xmlParser.html, retrieved 7/02/2012
[4] SAPI - www.msdn.microsoft.com/en-us/library/ee125663.aspx, retrieved 7/02/2012

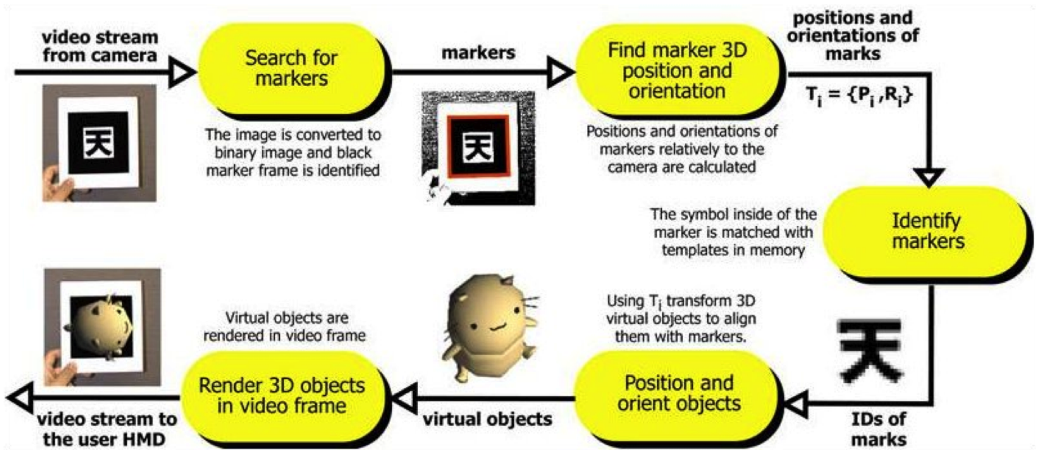


Figure 5.2: The ARToolkit tracking algorithm (Kato and Billinghurst, 1999).

components to the Intelligent Tutoring System, which allows it to analyze the user's behavior, provide feedback and make changes to the teaching approach as necessary.

The bulk of the work performed in the tracking module is handled by the underlying osgART software library. This package provides support for multiple camera-based tracking algorithms via the use of plug-ins in the form of dynamically-linked libraries. The details of each tracking approach are abstracted by the standardized plug-in interface, allowing the programmer to focus on higher-level aspects of building the Augmented Reality scene.

5.2.1 Tracking Algorithm

The tracking algorithm used with the motherboard assembly tutor is a variant of the ARToolkit marker-based approach (Kato and Billinghurst, 1999). As the description implies, this tracking algorithm utilizes two-dimensional fiducial markers that consist of black squares containing uniquely identifiable patterns. Figure 5.2 illustrates the algorithm procedure. The tracker analyzes video frames from a camera, searching for black squares. For each square that is found, the system examines the pattern inside of the square to determine whether it matches any of the patterns it is programmed to recognize. The tracker is told the exact size of each square ahead of time, so it is able to perform some complex calculations to determine the position and orientation of the marker relative to the camera with millimeter accuracy. The tracker is able to accurately detect and identify the squares even when they are at oblique angles relative to the camera.

The marker-based ARToolkit tracking approach was chosen for its performance and ease-of-use. Before deciding upon this algorithm, another approach was considered. This alternative solution is able to directly track the natural features of two-dimensional surfaces without the need for square markers. The natural feature algorithm requires greater processing power, but has the advantage of being more accurate and reliable in addition to being resistant to occlusion. Unlike with the marker-based ARToolkit tracker, the natural feature approach continues to track an object even if it is partially obscured, provided enough distinguishing features are visible.

Computer hardware components consist of mostly planar surfaces containing a variety of distinct visual features such as metallic circuitry and embedded logic units. It would have been advantageous to be able to track the components directly without the need for obtrusive markers. However, initial testing revealed that the motherboard and its components were in fact not flat enough to be considered two-dimensional surfaces for the purposes of the tracking system. The tracking worked well from an isometric perspective, but when the camera was tilted to a slight angle, the tracking performance became unacceptably poor.

Because the novel focus of this masters thesis is to explore the combination of Augmented Reality with Intelligent Tutoring Systems, rather than low-level Computer Vision and tracking, the marker-based ARToolkit approach was deemed sufficient for the purposes of the prototype, with the expectation that a superior tracking solution could be substituted in the future with minimal effort.

5.3 Display Module

The display module is responsible for everything the user sees through the head-mounted display. The HMD chosen for the project is a video-see-through device, meaning the user looks at a screen that displays a video reproduction of their first-person perspective via a camera attached to the front of the HMD. This is in contrast to optical see-through HMDs, with which the user views the world directly through a transparent medium onto which the virtual graphics are projected. Both approaches have advantages and disadvantages, but most consumer HMDs used for Augmented Reality applications are of the video-see-through variety. Please refer to section 5.5 for more details regarding the HMD used for this project.

As a result of this hardware choice, the first responsibility of the display module is to obtain video frames from the camera and draw them on the screen. After each frame is rendered, virtual graphics can be drawn on top of the video background in order to create the illusion that they exist within the real scene. All of the graphics produced by the display module are generated by the OpenSceneGraph[5] computer graphics library (OSG), which has been integrated into the osgART software package. OSG is based on the standard OpenGL[6] API, and provides a robust scene graph structure that sits on top of the basic OpenGL functionality. In addition to built-in support for materials, textures, lighting and shaders, OSG has a rich set of plug-ins that allow it to handle a wide variety of file formats for images, 3D models and sound. All of this makes it relatively quick and easy to bring in external resources and construct complex scenes.

5.3.1 Scene Graph Design

A scene graph is fundamentally a graph data structure that organizes the logical and spatial layout of a graphical scene. Like all trees in Computer Science, it contains nodes arranged in a hierarchical structure where each node can have many children but only one parent. Operations performed on parent nodes are automatically applied to their children, making it easy to organize complex series of transformations and effects for groups of objects in a scene.

Figure 5.3 provides a highly simplified illustration of the scene graph node structure that was created for the display module. Beneath the root node, the content is divided into three sub-trees with different rendering priorities representing the three layers of graphics that make up the Augmented Reality interface. The background layer, rendered first, consists of the raw video frames captured by the camera. The middle layer, rendered second, contains all of the spatially-registered 3D graphics that are overlaid on top of the video background and serve

[5] OpenSceneGraph - www.openscenegraph.org, retrieved 7/02/2012
[6] OpenGL - www.opengl.org, retrieved 7/02/2012

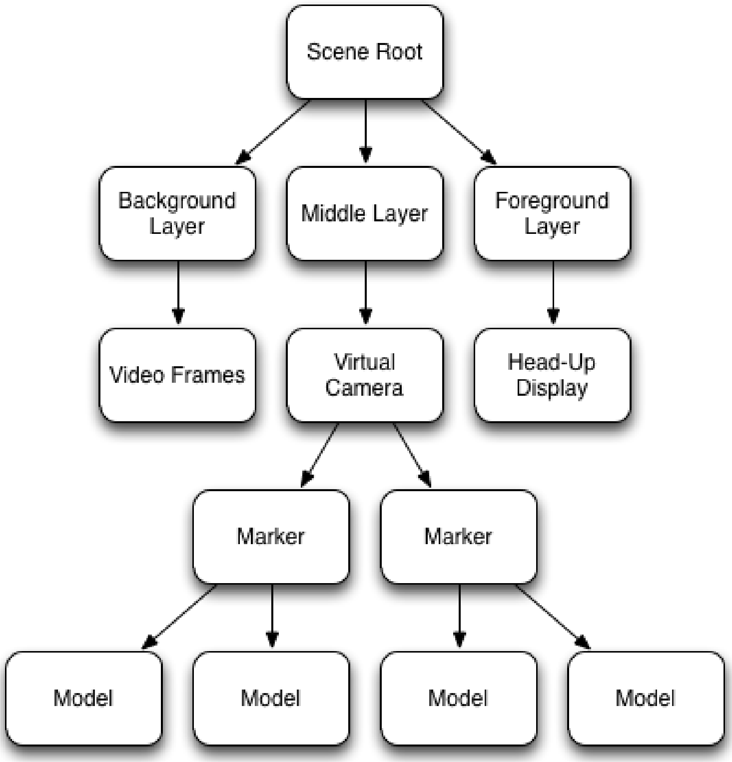


Figure 5.3: Simplified diagram of the scene graph node structure created for the display module.

as instructional cues for the user. The foreground layer, rendered last, contains all of the screen-aligned 2D graphics for the head-up display, which shows textual instructions and feedback from the Intelligent Tutoring System.

The majority of the work takes place in the middle layer. A graphical context is created in which a virtual viewpoint is placed at the origin of a three-dimensional coordinate system. The view from this virtual camera corresponds to that of the real camera affixed to the head-mounted display. The real camera captures the user's view of the real world, while the virtual camera captures the same perspective in the virtual world. The trick is to place the virtual objects in the correct positions on the coordinate system so they match up with corresponding objects in the real world. Then, when the two views are blended together, a convincing illusion is created.

Once the virtual camera has been initialized, a scene graph node is created for each of the fiducial markers used for tracking. For more details regarding the tracking approach, please refer to section 5.2. Each marker node encapsulates a 3D

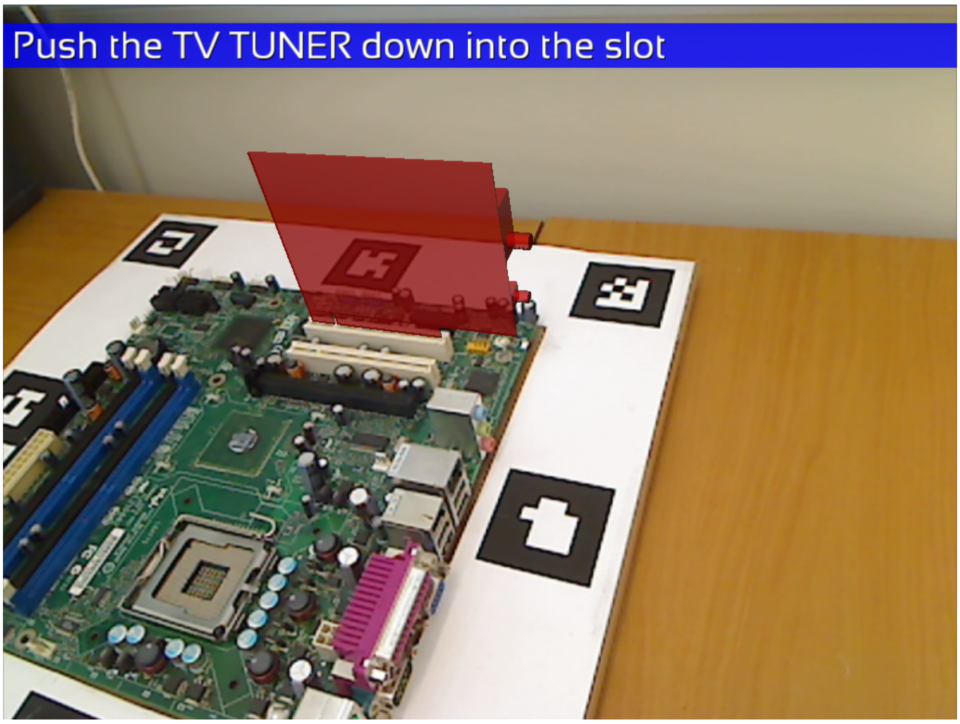


Figure 5.4: First-person view of the AR display for part of the TV tuner installation task. The red-colored 3D model indicates how the component should be inserted.

transformation matrix that represents the position and orientation of the marker relative to the camera. Because the views from the real and virtual cameras are superimposed, this transformation is valid for both the real marker and its corresponding representation in the virtual world. It is the job of the tracking system to constantly update the values of the virtual transformation matrices as the real markers change position relative to the camera.

Each marker node may have one or more graphical objects associated with it. In most cases these are 3D models, but there is also support for 2D images and sounds. Because the model nodes are children of the marker nodes, they inherit the transformations and are thereby "attached" to the markers. When the markers move relative to the camera, the tracking system updates the transformations and the models move accordingly. If a marker disappears from view, its children are hidden until the tracking system finds it again. Some 3D models may have animations associated with them, which are played automatically while the marker is in view. Figure 5.4 shows a first-person view of the display for part of the TV tuner installation task. The insertion animation is not visible in the picture.

A static scene that always remains the same would not be very useful for the purposes of an Augmented Reality tutor. The display module must be able to dynamically adjust the scene on the fly as the user progresses through the assembly or maintenance procedure. To accomplish this, the scene graph also contains "switch" nodes, which are used to hide and show groups of virtual content based on commands received from the Intelligent Tutoring System. Each step of the assembly process has one or more switch nodes associated with it, which are switched on when the task begins and switched off when the task is completed.

5.3.2 Creating 3D Models & Animations

Once the scene graph structure was designed, the virtual display content was created. The 3D Studio Max[7] graphic design application (produced by Autodesk) was used to generate accurate 3D models of the components to be installed on the computer motherboard, including memory, processor, graphics card, TV tuner card and heatsink. Models were also produced for relevant parts of the motherboard itself, such as the processor enclosure and memory securing mechanisms. Other 3D models, such as arrows, were created to assist with guiding the user through the tutoring process.

The models were then animated to illustrate the proper installation procedures. For example, the graphics card is visibly pushed downward into the PCI express slot, and the processor enclosure is opened before the processor is inserted. The animations were embedded into the exported 3D model files, which can be loaded directly into the display module by the appropriate plug-in in the OpenSceneGraph software library.

5.3.3 Head-up Display

A head-up display (HUD) is a screen-aligned graphical overlay that is not spatially registered within a scene. As the user looks around, the HUD components always stay in the same place on the screen, which is in contrast to the spatially-registered 3D models that are anchored to a position within the scene. In this case, the primary function of the HUD is to display messages from the Intelligent Tutoring

[7] 3D Studio Max - usa.autodesk.com/3ds-max/, retrieved 7/02/2012

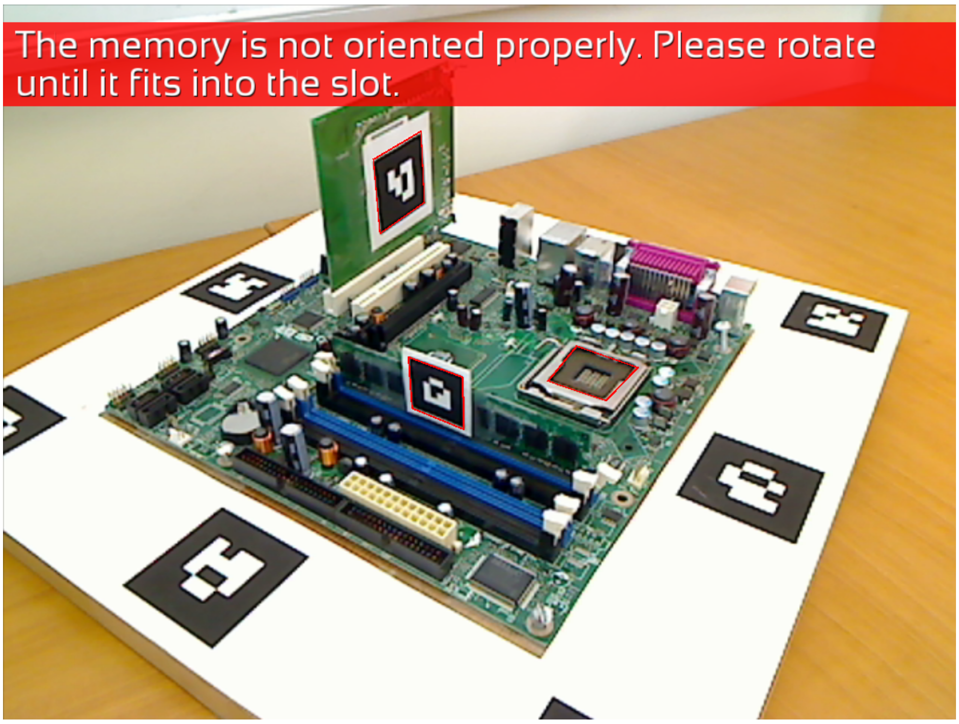


Figure 5.5: A negative feedback message for the memory installation task.

System. These textual messages include instructions for performing tasks as well as feedback about the student's performance. The text is displayed across the top of the screen and does not obstruct the user's view. The messages from the ITS fall into three categories—instructions, positive feedback (correct solution) and negative feedback (incorrect solution). The text is highlighted with a semi-transparent background that changes color based on the message type. Instructions are blue, positive feedback is green and negative feedback is red. Figure 5.5 illustrates an example of a negative feedback message for the memory installation task. In this case the memory module is in the correct position on the motherboard, but the orientation is incorrect because it has been inserted into the slot backwards. The ITS is able to distinguish between these conditions to provide specific feedback messages.

The HUD also utilizes text-to-speech technology to read the messages to the user. This is achieved via the Microsoft Speech API software library that is included with the Windows operating system. While the pronunciation is far from perfect, the spoken words certainly constitute a more natural form of interaction that improves the tutoring experience.

5.4 Communication Module

The communication module is responsible for relaying information between the AR interface and the Intelligent Tutoring System. The ITS controls what the user sees via the interface, and the interface tells the ITS what the user is doing. Because the ITS was created using the ASPIRE authoring system, it is a self-contained unit that resides on a dedicated server machine. This separation is advantageous for number of reasons. The interface and the ITS can be situated in completely different locations if necessary, and multiple interfaces can utilize the same ITS server simultaneously via the use of sessions.

Interaction with the ITS is achieved via remote procedure calls over a TCP/IP network connection. Remote procedure calls allow computer code on one machine to execute code on another machine. In this case, the arguments to the remote function are encoded as XML strings and sent to the server, which executes the function and similarly encodes the return values. The communication module utilizes a software library called XML-RPC for C++ on Windows to send and receive the XML strings. Another library called XMLParser is used to perform the actual encoding and decoding.

5.5 Hardware Setup

The hardware setup for the Augmented Reality interface consists of a head-mounted display, a camera, a computer running the Microsoft Windows operating system and the fiducial markers used for tracking. An Intel motherboard was selected for use with the computer assembly tutor prototype in addition to the five generic hardware components to be installed—memory, processor, graphics card, TV tuner card and heatsink. Figure 5.6 on the following page shows the hardware that was selected.

At least one unique marker was attached to each computer component to enable the tutor to identify and track its position. The motherboard itself was mounted on a sturdy wooden surface and surrounded with a configuration of eight separate markers. This group of markers works together with the tracking system to limit the effects of marker occlusion as users look around and move their arms

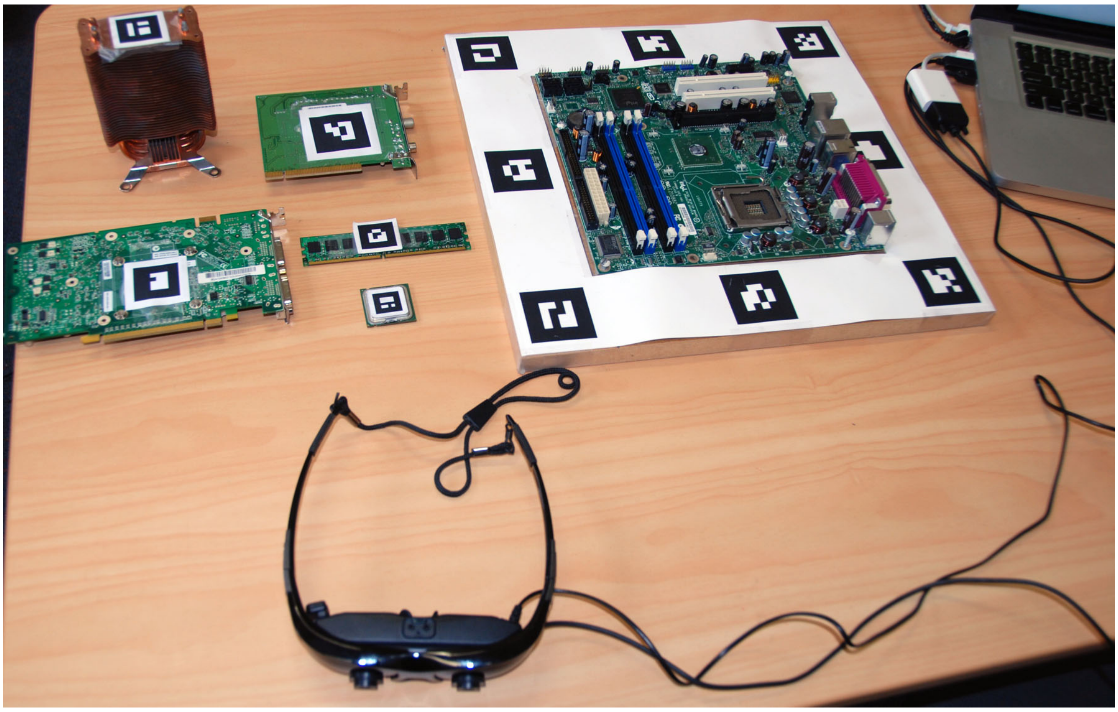


Figure 5.6: Hardware setup, including motherboard components, fiducial markers and HMD.

during the installation procedures. As long as the camera can see at least one of the eight markers, the tracking system is able to determine the relative position and orientation of the motherboard.

The HMD and camera combination chosen for the project is the Wrap 920AR model produced by Vuzix[8]. Currently retailing at $1500 USD with a similar form factor to normal sunglasses, this device represents the cutting edge of consumer-grade wearable display technology. The HMD supports a resolution of 1024x768 pixels and effectively simulates a 67-inch television as viewed from 10 feet with a 31-degree horizontal field of view. Stereoscopic viewing is supported with a number of 3D modes, and the front of the display is outfitted with two cameras for stereo video capture at 30 frames per second. The device connects to a computer via the standard VGA interface and also delivers audio via removable earbud headphones.

[8]Vuzix Wrap 920AR - http://www.vuzix.com/consumer/produces_wrap920ar.html, retrieved 7/02/2012

Chapter 6

Evaluation

This chapter covers the evaluation of the intelligent Augmented Reality tutoring system. We begin by discussing the evaluation goals and experiment design, and then present the results and conclusions that can be made. All documents used in the experiment can be found in Appendix A.

6.1 Evaluation Goals

The primary goal of the evaluation was to test the prototype motherboard assembly training system with a group of participants in order to answer the research questions outlined in Chapter 3. The most important of these asks how and to what extent Augmented Reality-based training can benefit from the use of intelligent tutoring approaches. To address this question, the evaluation compared the new intelligent Augmented Reality system with a more traditional AR tutor that does not employ an ITS. Another important question involves examining the difference in knowledge retention between the traditional and intelligent training approaches, compared to only measuring performance while using the AR system as previous studies have done. To accomplish this, the evaluation was split into two phases—a training phase using the tutor, and a testing phase (without the tutor) that measured the extent to which they retained the knowledge and physical skills they acquired.

Table 6.1: Feature comparison between the intelligent and traditional AR tutors.

Tutor Type	AR Content	HUD	Audio	Feedback	Customized Instruction
Intelligent	Yes	Yes	Yes	Yes	Yes
Traditional	Yes	Yes	Yes	No	No

6.2 Intelligent Tutor vs. Traditional Tutor

In order to evaluate the effectiveness of the intelligent approach, a traditional AR training system was created for comparison. In order to isolate the intelligence factor, the systems must be identical in every way except for the features related to the ITS. This was relatively straightforward to achieve in the case of the motherboard assembly tutor. Because the ITS is a completely separate module from the AR interface, it was possible to remove this connection and create a version of the tutor that proceeds blindly through the assembly steps like slides in a slide show. This traditional system is like other existing AR assembly tutors in that it does not customize the experience by paying attention to what the student is doing and providing feedback—it simply shows the student what needs to be done for each step and moves on. In addition, with the intelligent system, the ITS controls the ordering of the assembly steps and can make decisions about what material to present next based on the student's performance. The ordering of the steps in the traditional system is fixed. The tutors have the same interface and provide the same visual and oral instructions for each step, so the only differences lie in the features directly related to the ITS. Table 6.1 summarizes the differences between the intelligent and traditional tutors.

6.3 Experiment Design

This section discusses the experimental design, including a formal definition of the hypothesis, a description of each phase of the experiment and an explicit outline of the procedure.

6.3.1 Hypothesis

The experiment is designed to address the following overarching hypothesis: **The use of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks can significantly improve the learning outcome over traditional AR approaches that do not employ an ITS.** There are a variety of sub-hypotheses that will be used to confirm or refute this conclusion in the analysis of the results (Section 6.5).

6.3.2 Description

During the experiment, the participants were split between two independent between-subjects conditions. One group used the intelligent AR motherboard assembly tutor, while the other group used the traditional AR tutor. Great care was taken to select people who had minimal prior experience with computer hardware assembly. To measure this factor, all participants were given a written pre-test asking them to identify the five hardware components as well as indicate where they are installed on the motherboard. In addition, on the post-experiment questionnaire, the participants were asked to rate their own prior hardware experience on a scale from one (not experienced) to seven (very experienced).

Following the written pre-test, the participants were given an orientation to the AR tutor (intelligent or traditional) and its operation procedures. After they put on the head-mounted display, the tutor guided them through the process of identifying and installing five motherboard components—memory, processor, graphics card, TV tuner card and heatsink. After all of the components were assembled, the tutoring phase was complete and the participants were given a written post-test that was similar to the pre-test to measure how well they learned from the tutor. The two written tests covered the same material, but were not identical.

Following the written post-test, the participants were asked to perform a physical post-test in which they attempted to assemble the motherboard components once more—this time without the help of the tutor. The purpose of this phase was to measure how well they retained the physical assembly knowledge gained from

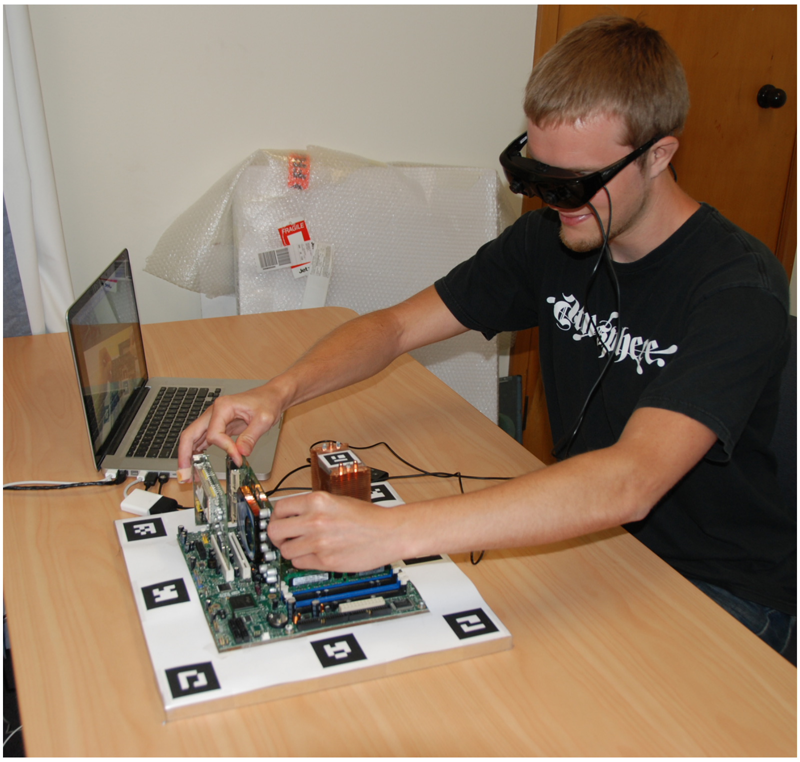


Figure 6.1: Participant assembles the motherboard with the help of the AR tutor.

the tutoring process. Given only the name of each component, the participants had to correctly identify and install them one by one. In addition to qualitative observations, a number of quantitative measures were taken during this process, including task completion time and error counts.

After the physical post-test, the participants completed a questionnaire, which prompted them to provide detailed feedback regarding their experience with the tutor. In addition to asking about prior hardware experience, the questionnaire contained a variety of questions with Likert-scale ratings. These asked the participants to indicate whether they thought the tutor was effective, whether they were satisfied with the 3D AR content, whether they thought the AR training system was more effective than other types of media such as instructional videos or paper manuals, and whether they felt physically or mentally stressed during the tutoring process. Participants also had the opportunity to provide additional written feedback in response to the questions.

6.3.3 Procedure

This section provides a concise outline of the experiment procedure.

- **Introduction** - Participant is given an information sheet and completes a consent form.
- **Written Pre-test** - Tests prior computer hardware knowledge.
- **Orientation** - Participant is introduced to the hardware and learns how to use the tutor (intelligent or traditional).
- **Tutoring** - The tutor guides the participant to install the components.
- **Written Post-test** - Tests knowledge gained from the tutor.
- **Physical Post-test** - Tests physical skills gained from the tutor.
- **Questionnaire** - Subjective feedback.

6.4 Quantitative Results

A total of 16 people participated in the experiment. They were divided into two even groups—eight participants for the intelligent AR tutor and eight for the traditional AR tutor. This is a relatively low number for a between-subjects test, but was deemed sufficient to satisfy the statistical requirements. All of the participants were university students between the ages of 18 and 45. Because the participants were divided into two groups (intelligent tutor and traditional tutor), a series of paired and unpaired t-tests provided sufficient statistical analysis. In general, conducting multiple statistical tests on the same data set necessitates the use of Bonferroni correction, even when the conclusions supported by the tests are independent of each other. Bonferroni correction greatly reduces the probability of a Type 1 statistical error (false positive), so the p-value of each of the following hypotheses is analyzed both with respect to the typical $\alpha = .05$ (95% confidence) and a corrected value of $\alpha/n = .0083$ where $n = 6$ for the 6 tests being performed.

6.4.1 Written Pre-test and Post-test

Measured Results

Table 6.2 on the next page summarizes the written pre-test and post-test scores for the intelligent and traditional tutor groups. The maximum possible score for both tests was 10 points. The pre-test result for the intelligent tutor group was a mean score of 2.50 points with a standard deviation of 2.27. The post-test resulted

Table 6.2: Written pre-test and post-test scores for the intelligent and traditional tutor groups. The maximum possible score was 10 points.

Group	Pre-test		Post-test		Normalized Gain	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Intelligent	2.50	2.27	9.13	1.13	0.66	0.26
Traditional	2.63	1.92	6.63	1.77	0.40	0.21

in a mean score of 9.13 points with a standard deviation of 1.13. The mean of the normalized gains (post-test score minus pre-test score for each participant, divided by the maximum possible score) was 0.66 with a standard deviation of 0.21.

The pre-test result for the traditional tutor group was a mean score of 2.63 points with a standard deviation of 1.92. The post-test resulted in a mean score of 6.63 points with a standard deviation of 1.77. The mean of the normalized gains for this group was 0.40 with a standard deviation of 0.21.

t-Test Results

It is important to establish whether the participants in each group had a similar level of knowledge prior to using the tutor and that one group did not have a significant advantage over the other. A two-tailed heteroscedastic t-test with two-sample unequal variance was performed on the pre-test scores of both groups, yielding a p-value of .907. This value is significantly larger than the α value of .05, which means we do not reject the null hypothesis. This implies that there was no significant difference in pre-test scores between the groups. In fact, because the p-value is so large, we can be reasonably confident that a Type 2 statistical error has not occurred (false negative). It does not make sense to apply Bonferroni correction with this test because Bonferroni correction only protects against Type 1 errors (false positives).

To determine whether a significant level of learning actually took place with each tutor, a two-tailed paired t-test was conducted on the pre-test and post-test scores within each group, yielding p-values of .000183 and .00113 for the intelligent and traditional groups respectively. Both of these values are significantly less than the desired α value of .05, which results in a firm rejection of the null hypotheses.

This implies that there was a significant difference between pre-test and post-test scores within each group. Furthermore, the p-values are both less than the Bonferroni-corrected value of .0083 (.05/6), which makes an even stronger case for the effectiveness of both tutors.

It is also important to determine the value of the intelligent AR training approach. A two-tailed heteroscedastic t-test with two-sample unequal variance was performed on the post-test scores of both groups, yielding a p-value of .00561. This is significantly less than the desired α value of .05, which results in a firm rejection of the null hypothesis. This implies that there was a significant difference in post-test scores between the groups. Furthermore, this p-value is less than the Bonferroni-corrected value of .0083 (.05/6), which makes a very strong case for the superiority of the intelligent AR tutor.

A two-tailed heteroscedastic t-test with two-sample unequal variance was performed on the normalized learning gains of both groups, yielding a p-value of .0460. This is slightly less than the desired α value of .05, which results in a successful rejection of the null hypothesis. This makes an even stronger case that the intelligent tutor provides a superior learning outcome to the traditional approach. This p-value is greater than the Bonferroni-corrected value of .0083 (.05/6), however, which weakens the argument.

6.4.2 Physical Post-test

Measured Results

The quantitative measures for the physical post-test consisted of the number of errors made and the total completion time to install all five motherboard components. Table 6.3 on the following page summarizes the results. The mean physical post-test completion time for the intelligent tutor group was 56.56 seconds with a standard deviation of 11.31. The mean error count was 0.50 with a standard deviation of 0.93. The errors generally fit into two categories—failing to match the name with the correct component, or incorrectly performing the installation procedure.

The mean physical post-test completion time for the traditional tutor group

Table 6.3: Total completion time (seconds) and number of errors made during the physical post-test for the intelligent and traditional tutor groups.

Group	Time		Errors	
	Mean	**Std Dev**	**Mean**	**Std Dev**
Intelligent	56.56	11.31	0.50	0.93
Traditional	81.13	21.11	1.00	0.93

was 81.13 seconds with a standard deviation of 21.11. The mean error count was 1.0 with a standard deviation of 0.93.

t-Test Results

A two-tailed heteroscedastic t-test with two-sample unequal variance was performed on physical post-test completion times between the groups, yielding a p-value of .0148. This is significantly less than the desired α value of .05, which results in a successful rejection of the null hypothesis. This implies that there was a significant difference in physical post-test completion times between the groups. The p-value is greater than the Bonferroni-corrected value of .0083 (.05/6), which weakens the conclusion.

A two-tailed heteroscedastic t-test with two-sample unequal variance was performed on the error counts for both groups, yielding a p-value of .298. This is greater than the desired α value of .05, which does not allow for a rejection of the null hypothesis. This implies that there was no significant difference in physical post-test error counts between the groups.

6.4.3 Questionnaire

The quantitative portion of the questionnaire consisted of 10 Likert-scale ratings, which prompted participants to indicate their prior computer hardware experience level along with their opinions regarding their performance of the tutor. All of the Likert-scales ranged from one to seven, and the results are summarized in table 6.4. For more information about the individual questions, please see the questionnaire in Appendix A.5.

Table 6.4: Mean Likert-scale responses (ranging from 1 to 7) for the questionnaire. The question numbers in the table correspond with those on the questionnaire.

Group	Q1	Q2	Q3	Q4	Q6	Q8	Q9	Q10	Q11	Q12
Intelligent	3.4	6.0	6.2	6.1	6.1	5.8	5.6	6.5	1.4	1.6
Traditional	2.1	6.0	6.4	6.1	5.5	6.5	5.5	6.8	1.9	1.8

6.5 Analysis

This section provides an analysis of the experiment and its outcomes, beginning with statistically-supported conclusions drawn from the quantitative results, moving on to examine threats to internal and external validity and finally discussing pertinent qualitative observations and feedback from the questionnaire.

6.5.1 Statistical Discussion

As discussed in section 6.3.1, the experiment was primarily designed to address the following hypothesis: **The use of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks significantly improves the learning outcome over traditional AR approaches that do not employ ITSs.** A number of statistically-supported sub-hypotheses are listed here in order to qualify this general statement. Please see the quantitative results (section 6.4) for specific details regarding the statistical tests performed.

There was no significant difference in written pre-test scores between the traditional and intelligent tutor groups. This conclusion implies that the participants in each group had a similar level of knowledge prior to using the tutors and that one group did not have a significant starting advantage over the other.

The written post-test scores were significantly greater than the pre-test scores for both groups. This conclusion implies that both the intelligent and traditional tutors were successful at teaching the motherboard assembly procedures. A significant level of learning took place with both tutors, which makes a solid argument for the effectiveness of AR training in general.

The written post-test scores for the intelligent tutor group were significantly greater than the post-test scores of the traditional tutor group.

This conclusion is essential to demonstrating the value of the intelligent AR training approach. Participants who used the intelligent tutor were able to learn and retain knowledge more effectively than those who used the traditional tutor.

The normalized learning gains (post-test score minus pre-test score for each participant, divided by the maximum possible score) were significantly greater for the intelligent tutor group.

This conclusion makes an even stronger case that the intelligent tutor provides a superior learning outcome to the traditional approach. The gain is a measurement of learning improvement on per-student basis, which more directly examines the impact of the tutoring experience. It is more difficult to obtain a significant difference between groups using this metric, which is why it makes for a stronger argument.

The Effect Size for the normalized gains was calculated to be 0.981, which implies an expected test score improvement of nearly one standard deviation for students using the intelligent tutor over those who use the traditional tutor.

The Effect Size is determined by computing the difference of the means of the normalized gains for the two groups and dividing by the pooled standard deviation of all of the gains for both groups. This value, calculated to be (0.663-0.40)/0.268 = 0.981, suggests that the improvement in test scores as a result of using the intelligent tutor instead of the traditional tutor is expected to be equal to nearly one standard deviation. This is a significant improvement.

The task completion times for the physical post-test were significantly less for the intelligent tutor group.

This conclusion demonstrates that participants who used the intelligent tutor completed the installation tasks in the physical post-test faster and more confidently than those who used the traditional tutor. This suggests that the intelligent tutor was not only more effective at instilling theoretical knowledge, but also more effective at teaching the physical skills.

While the participants in the intelligent tutor group made fewer errors during the physical post-test than those in the traditional tutor group, the difference was not found to be statistically significant.

Unfortunately it was not possible to conclude that there was a statistically significant difference in error rates between the groups. A successful rejection of the null hypothesis would have made a strong argument that participants who used the intelligent tutor were able to retain knowledge of the correct physical installation procedures to a significantly greater extent than those who used the traditional tutor.

6.5.2 Questionnaire Likert-scales

None of the Likert-scale ratings on the questionnaire yielded statistically significant differences between the intelligent and traditional tutor groups, but the opinions expressed are still interesting as they pertain to both AR tutors. The Likert-scale answers support the following conclusions. The mean and standard deviation of the responses for each question can be found in section 6.4.3.

The participants generally had little to no experience with computer hardware assembly procedures prior to the study.

44% of the participants had no prior experience at all, while 69% rated themselves three or lower on a seven-point scale from “not very experienced” to “very experienced”. None of the participants rated themselves "very experienced".

The participants from both the intelligent and traditional groups felt strongly that the tutor was effective and that they were successfully able to learn the assembly tasks.

81% of the participants rated their level of agreement at six or higher out of seven with the statement that they were successfully able to learn the motherboard installation procedures. 81% also rated the effectiveness of the tutor at six or higher. None of the participants felt that they were completely unsuccessful or that the tutor was ineffective.

The participants felt strongly that the Augmented Reality tutoring approach was a more effective method of teaching motherboard assembly tasks then reading an instructional manual, and, to a lesser extent, more effective than an instructional video.

81% of the participants rated their level of agreement at six or higher out of seven with the statement that the AR training approach would be more effective than reading a paper manual. 56% had the same level of agreement with the statement that the AR training approach would be more effective than watching an instructional video.

The participants found the Augmented Reality tutors to be very interesting, and they experienced low levels of mental and physical stress during the tutoring process.

94% of the participants rated their interest level with the AR tutor at six or higher out of seven. 88% of participants rated their mental stress level at two or lower out of seven, while 81% rated their physical stress level at two or lower.

6.5.3 Qualitative Observations & Feedback

There are a variety of interesting qualitative results stemming from experimenter observations and written questionnaire answers.

Qualitative Feedback

The written feedback on the questionnaire was generally quite positive for both the intelligent and traditional AR tutors. Most participants felt that the visual step-by-step instructions were very helpful—the tutors allowed them to proceed at their own pace without the overhead of stopping and starting a video or interpreting written diagrams. The immersive first-person experience provided by the head-mounted display was engaging, and the system as a whole was interesting and fun to use. Some of these responses can be attributed to the novelty factor associated with Augmented Reality, but the fact remains that the participants generally found the tutors to be both effective and entertaining. Many of the participants in the intelligent tutor group felt that the feedback provided by the tutor in response to their solutions was very helpful in terms of confirming whether they were right or wrong. This additional support more closely simulated a human teacher, the result of which was reflected positively in the written questionnaire responses.

In addition to the positive feedback, there were a number of constructive criticisms. Although the Vuzix Wrap 920AR head-mounted display represents the latest in consumer HMD technology, the participants still found it to be somewhat uncomfortable to wear. This issue was exacerbated by the fact that the display was operated in monoscopic mode rather than stereoscopic, which resulted in a lack of depth perception. As a result, participants sometimes found themselves using the display to understand what needed to be done for each step, and then directing their gaze beneath the display when actually performing the procedure. A more comfortable display would eliminate the need for this context switching and would improve the immersive quality of the experience.

Another criticism stemmed from the fact that the textual instructions were screen-aligned in typical HUD fashion. Reading the text required the participants to shift their focus from looking into the scene to looking at the text displayed on the surface of the screen. It may have been more natural to use spatially-registered text that appeared within the scene to keep the students immersed in the mixed-reality environment. Other criticisms addressed the graphical glitches that resulted from poor tracking performance. The virtual content would sometimes "jiggle" or disappear entirely when the tracking system was unable to obtain enough information about the markers. These issues could be addressed by substituting a more robust tracking approach—perhaps one that utilizes multiple cameras and directly tracks the natural features of the motherboard components without the need for markers.

Experimenter Observations

While determining the correct position of the components was relatively easy, the participants sometimes had difficulty determining the proper orientation. This was partially due to a lack of orientation cues in some of the virtual content shown via the head-mounted display. The memory and processor are essentially symmetrical in shape, and it can be difficult to determine which direction the virtual rendering is facing when there are no distinguishing features indicating which side is which. In these cases, It would be helpful to have some additional AR cues to help the student infer the correct orientation. One idea would be to attach virtual arrows to the motherboard slot as well as the actual component to be inserted, prompting the student to line up the arrows with each other.

When this type of orientation mistake occurred, the intelligent AR tutor was able to detect the error and inform the student that the orientation was incorrect. The participant was required to correct the mistake before being allowed to proceed. The traditional tutor by nature was unable to observe or correct errors, and they often went unnoticed by the student. In these cases, the student typically made similar mistakes during the post-test. This supports the claim that the added feedback from the Intelligent Tutoring System improved the learning outcome over the traditional AR training approach, particularly in situations where it is easy to make a mistake.

There were a few instances where the Intelligent Tutoring System was a detriment to the learning process. Due to the challenge of robustly tracking the motherboard and its components in 3D space, there were times when the tracking system relayed inaccurate information to the ITS. This often resulted in false-negative responses where the system would tell the student that the solution for a given assembly step was incorrect when in fact no errors had been made. In practice, the participants were able to infer that the system had made a mistake in judging the solution, but these mishaps were certainly not beneficial to the learning process. The traditional tutor by nature did not examine the solutions and thus did not have this problem.

6.6 Threats to Validity

There are a number of potential threats to the validity of the experiment. These threats have been divided into internal and external sources and are addressed in the following subsections.

6.6.1 Internal Threats

Internal validity addresses the true causes of the outcomes of an experiment. Strong internal validity means the independent and dependent variables are measured reliably, and also that there is a strong justification that causally links the independent variables to the dependent variables. In this experiment, the independent variable was the was the tutor group (intelligent or traditional) to which each participant belonged, while the dependent variables were the written pre-test and post-test scores along with the completion time and error rates for the physical post-test. Due to the straightforward nature of the data collection methods, it is likely that the variables were measured reliably, but there are a number of other valid concerns relating to the causal link between the tutor groups and the measured results.

Unaccounted Subject-Related Factors Between Groups

The fact that there was no significant difference in pre-test scores between the two groups suggests that the outcomes of the experiment were not a result of a difference in prior computer hardware experience. In addition, great care was taken to balance the groups in terms of other subject-related variables including gender and age. However, it is always possible that there was an unbalanced subject-related variable that contributed to the difference in post-test results, such as ethnicity, socioeconomic status or general mental ability. The fact that several of the participants did not learn English as their first language could have affected their ability to learn the names of components and follow instructions.

Repeated Testing

Repeated testing is another potential threat to internal validity. Because the participants were given two written tests on the same material at two different points in time, it is possible that there were some learning effects that resulted from the testing procedure itself rather than from the independent variable (tutor). Although the tests were not identical, it is possible that some of the information gained from the pre-test allowed the participants to perform better on the post-test. However, because the testing procedure was identical for both tutor groups, it is reasonable to conclude that the repeated testing factor did not contribute to the significant difference in learning outcome between the groups.

6.6.2 External Threats

External validity refers to the ability to generalize an experiment to other people and situations. The population from which the participants were selected is fairly well-defined: University students who have minimal prior experience with computer hardware. Due to the diversity found in the university environment, there were a variety of ages (18 to 45) and nationalities represented, including New Zealand, the Netherlands, Brazil, Iraq, Malaysia and the United States. Despite this diversity, however, there are a number of concerns that could be raised when making general conclusions from this group.

Firstly, the experiment explicitly called for participants who had minimal prior experience with computer hardware in order to maximize the learning potential for the tutors. Correspondingly, the resulting conclusions should be restricted to inexperienced populations in order to maintain strong external validity. It is possible that different effects would be observed with a more experienced group—perhaps the additional overhead involved with the intelligent AR tutor ultimately becomes a detriment when the student is already familiar with the assembly process.

Another concern stems from the fact that all of the participants were well-educated adults over the age of 18. It is possible that the results of the experiment would have been different given a population of children, seniors or less-educated individuals. Perhaps younger people would not react as well as adults to the structured feedback provided by the intelligent tutor, faring better by discovering the correct position and orientation of the components on their own. They may also have greater difficulty completing a written test of their knowledge, which could affect the results. Once again, it is necessary to constrain the conclusions of the study in order to maintain strong external validity—in this case by applying them only to well-educated adults.

6.7 Conclusions

The goal of the experiment was to evaluate the effectiveness of the intelligent Augmented Reality tutor in teaching the motherboard assembly procedures. To achieve this goal, a traditional AR training system was created for comparison. The two tutors were identical in every way except for features directly related to the Intelligent Tutoring System.

After testing both tutors with a population of adult university students, the intelligent AR tutor was found to be significantly more effective than the traditional AR system. Participants who used the intelligent tutor scored an average of 25% higher on a written post-test of the teaching material than those who use the traditional tutor, which was found to be a statistically significant improvement. There was no significant difference in pre-test scores or other subject-related factors between the two groups, so the improvement can be attributed directly to the superiority of the intelligent AR training approach. Furthermore, the average learning gain, measured by subtracting each participant's pre-test score from the post-test score, was significantly greater for the intelligent tutor group, which strengthens the argument. To further generalize this conclusion, the effect size of the difference in learning gains was calculated to be 0.981. This implies that the expected increase in test scores as a result of using the intelligent AR tutor instead of the traditional AR tutor is equal to nearly one standard deviation, which is a significant improvement.

The participants also completed a physical post-test of their newly acquired motherboard assembly skills. Those in the intelligent tutor group assembled the components an average of 30% faster, which was a significant difference. The intelligent tutor group also made fewer errors, although the difference in error counts was not found to be statistically significant.

The results of the experiment confirm the overarching hypothesis that the use of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks significantly improves the learning outcome over traditional AR approaches that do not employ ITSs, particularly in situations where mistakes are likely to occur.

Chapter 7

Conclusion

7.1 Summary

Like many emerging technologies, Augmented Reality has gradually made its way from military applications to academic laboratory prototypes and is now becoming mainstream among the general public. Most consumer-level applications currently have limited usefulness, but as AR and Computer Vision continue to mature, there will be an increasing variety of practical applications that impact everyday life in truly meaningful ways.

One area that holds particular promise is the realm of education and training, where AR has been shown to improve learning through visualization and interactivity. Computers have long been utilized for teaching purposes, and Intelligent Tutoring Systems represent the latest evolution of Artificial Intelligence for education. In a similar fashion to AR, ITSs have matured and entered a realm of practicality in which they can be deployed successfully in real classroom situations. All of this begs the question of whether the distinct fields of AR and ITSs can benefit from each other when applied together in an educational context, and it is this question that provides the inspiration for this masters thesis project. While there has been a significant amount of research exploring the combination of ITSs with Virtual Reality, there is very little prior research investigating their integration with AR.

The primary goal of the project was to create an Augmented Reality training system that utilizes an Intelligent Tutoring System to provide a robust and customized learning experience for each user. To demonstrate the system, a prototype application was created that teaches users how to assemble hardware components on a computer motherboard. Due to a lack of prior research investigating the combination of the two fields, the first task was to determine the key properties any ITS should have in order to work well with an AR interface in order to teach a physical assembly task. After outlining a list of desired properties, seven existing ITS authoring solutions were examined with respect to these characteristics. The clear winner was ASPIRE, which is the constraint-based authoring system developed by the Intelligent Computer Tutoring Group at the University of Canterbury (Mitrovic et al., 2008). The next task was to use the authoring interface in ASPIRE to create the ITS back-end that controls the training process. After the ITS was completed, the software and hardware components of the AR interface were developed, and the two modules were connected to create a working system.

To evaluate the intelligent AR training approach, a traditional system was created for comparison that was identical in every way except for features relating directly to the ITS. The results of the evaluation revealed that participants who used the intelligent system surpassed those who used the traditional system by an average of 25% on written tests and also significantly outperformed them on physical tests of their knowledge. On average, the intelligent tutor group completed the physical test 30% faster than the traditional group while also making fewer mistakes. These results support the conclusion that the combination of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks can significantly improve the learning outcome over traditional AR approaches that do not employ ITSs. Furthermore, the usefulness of the ITS appears to be directly related to the complexity of the task. For very simple tasks, the student is less likely to make a mistake, and thus the ITS does not greatly influence the learning process. However, for more complex or open-ended tasks, the student makes more mistakes and the robust scaffolding and feedback provided by the ITS has a greater impact.

7.2 Contributions

The following research contributions have been made upon completion of this masters thesis project:

- Answers to the general research questions presented in section 3.1 (restated below).
- A prototype system that combines Augmented Reality with Intelligent Tutoring Systems to teach computer motherboard assembly.
- An evaluation of the prototype, including a formal user study, which substantiates the research conclusions.
- A modular software architecture that can be reused with other projects involving intelligent AR-assisted training for assembly and maintenance.
- A full written report detailing all aspects of the research.

How and to what extent can Augmented Reality-based training benefit from the use of intelligent tutoring approaches?

The use of Intelligent Tutoring Systems with Augmented Reality training for assembly and maintenance tasks can significantly improve the learning outcome over traditional AR approaches that do not employ ITSs, particularly in situations where mistakes are likely to occur.

What type of Intelligent Tutoring System is best suited for use with Augmented Reality-assisted assembly and maintenance tasks?

As discussed in section 4.1, the ideal ITS has flexible communication methods, supports both procedural and non-procedural tasks, accepts multiple solutions to problems, employs dynamic teaching strategies and utilizes an adaptive student model to customize the learning experience.

Can intelligent Augmented Reality-based training enable users to learn and retain assembly and maintenance skills more effectively when compared with traditional AR training approaches?

As discussed in section 6.7, the results of the evaluation indicate that participants who used the intelligent AR tutor were able to learn and retain the motherboard assembly skills significantly more effectively than those who use the traditional AR tutor.

7.3 Challenges & Lessons Learned

There are a variety of challenges inherent to the combination of AR with ITSs. The biggest issue is the fact that tracking demands of an intelligent AR system are significantly greater than those of more traditional AR training approaches that do not use ITSs. This is because the tracking is not only used for displaying the virtual instructions, but also to inform the system about what the student is doing so that it can customize the tutoring process and provide detailed feedback. This is much easier to achieve with Virtual Reality, which could be why there has been a significant amount of research exploring the combination of VR and ITSs, but very little research examining AR and ITSs. With VR, the user interacts only with virtual objects, and the system automatically knows everything about their states and positions in the world. When the user picks up a virtual coffee cup, the system must track the user's movements, but there is no need to track the cup itself because it is part of the virtual environment and the system inherently knows its location. With AR, the user can interact with real objects, so the system must be able to directly track the real coffee cup and distinguish it from other things in the world. When the user throws the coffee cup across the room, the system must be able to identify it and determine its exact position and orientation. This clearly poses a greater challenge than the VR case.

Another significant challenge also stems from the issue of tracking. The usefulness of an ITS is related to the difficulty of the material it is teaching. If the student is learning something very simple, the intelligence of the tutor doesn't matter as much because the student will learn the material easily regardless of the method. However, with more complex material where it is easier to make mistakes, the ITS becomes more useful due to its interactivity and its ability to customize the learning experience. Unfortunately, when it comes to AR training, more complex tasks generally require more robust tracking. For example, fixing a car engine is a more complex task with greater tracking requirements than assembling components on a computer motherboard. When designing an intelligent AR training system, the task must be complex enough for the ITS to be useful, but not so complex that the tracking becomes too difficult.

For this reason, computer motherboard assembly is not an ideal task as far as demonstrating the usefulness of intelligent Augmented Reality training. It is a great choice from a purely AR perspective because the components are mostly planar objects that are easy to track, and they are installed into clearly-defined positions on the motherboard. The installation procedures themselves are relatively simple, and the design of the connectors intentionally makes it difficult to insert components in the wrong positions or orientations. In fact, it is often possible to deduce where a component is installed by simply examining the connector and matching it with a corresponding slot on the motherboard. This design makes it less likely for the student to make mistakes, which marginalizes the usefulness of the ITS. However, there were a few specific instances where errors were more likely, such as inserting the processor in the correct orientation. The ITS demonstrated its usefulness in these cases by successfully correcting mistakes. Despite lacking an ideal training task, the intelligent AR training system was shown to significantly improve the learning outcome over the traditional approach, which bodes very well for future applications in more complex scenarios.

Another challenge stems from the fact that the primary focus of this intelligent AR project is on learning rather than performance. Prior AR training research tends to focus on improving user efficiency while using the system as opposed to measuring how effectively the system teaches and imparts knowledge. This is likely due in part to the fact that education and learning are very complex topics with many qualitative and subjective factors, which makes it difficult to quantify and compare approaches. It is fairly straightforward to analyze objective performance measures such as completion time and error rates, but measuring learning and knowledge retention is a much more formidable task that requires deeper analysis and interpretation. While this poses a greater challenge to the researcher, the resulting conclusions are arguably more meaningful in terms of addressing the underlying goal of training.

7.4 Future Work

There are many future research directions to be explored in the realm of intelligent Augmented Reality. Specifically, the system developed for this masters thesis project can be extended in a number of tangible ways. One idea involves integrating a virtual character into the AR tutoring environment. Virtual characters are visual representations of computer entities that are typically capable of speaking, moving and expressing qualities of living beings such as emotions and body language. Research has shown that virtual characters can be beneficial in tutoring situations by more closely simulating a human teacher (Liu and Pan, 2005). The student becomes more engaged and invested in the tutoring experience, which can result in an improved learning outcome. One can imagine a motherboard assembly tutor that places a 3D virtual character into the environment with the student. The character would essentially allow the ITS to inhabit the world with the user, where it could give verbal instructions, make physical gestures and demonstrate installation procedures like a human teacher would. The character could be equipped with speech recognition and language processing capabilities that would allow it to respond to verbal questions or prompts, enabling more natural interaction between the student and the ITS.

It would also be worthwhile to conduct more evaluations to further examine the impact of ITSs on AR training. For example, the participants in this study were tested on their knowledge immediately after using the AR tutor, but another study could look at long-term retention rates over multiple sessions to further refine the tutoring approach. In addition, other studies could look at combining ITSs with AR to teach more abstract topics, such as biology or history. While AR has already been applied to some of these educational domains, the existing systems do not employ ITSs to teach the concepts.

Another area for future study lies in the realm of mobile AR. While the concept of AR on portable devices is not new, there are very few (if any) mobile AR training systems that utilize an ITS to provide a customized learning experience. One can imagine a future where it is possible to download an application onto a smartphone that diagnoses car problems and teaches the user to change the oil filter using intelligent AR.

Tracking is another area in which the intelligent AR prototype can be improved. The current solution utilizes a fiducial marker-based approach, which has a number of drawbacks including limited accuracy, poor resistance to occlusion and the distracting nature of the markers themselves. There are a number of better tracking approaches that directly observe the natural features of objects such as colors and textures without the need for obtrusive markers (Neumann and You, 1999). There are also tracking solutions that utilize multiple cameras to reduce the effect of occlusion. Some of the more advanced optical approaches use stereoscopic cameras and depth mapping to determine the three-dimensional shapes of objects (Vacchetti et al., 2004). This 3D tracking allows the system to generate a model of the environment on the fly, which would enable it to adapt to new scenarios such as different brands of computer motherboards and components. Substituting a superior tracking solution into the existing intelligent AR prototype would vastly improve the motherboard training experience in addition to allowing support for more complex training tasks that require more robust tracking.

Equipped with improved tracking and tutor logic, the intelligent AR approach could be used to teach people how to drive a car, assemble furniture from the store or make gourmet lasagna in the kitchen. One can imagine a sports training tutor that helps users improve their golf swing, or a system that teaches how to build a radio using parts from a hardware store. Learning new practical skills could become easier and more accessible with the combined power of AR and ITSs, ushering in a new era of human growth and achievement.

Bibliography

V. Aleven, J. Sewall, B. M. McLaren, and K. R. Koedinger, “Rapid authoring of intelligent tutors for real-world and experimental use,” *Sixth International Conference on Advanced Learning Technologies*, pp. 847–851, Jun 2006.

J. Anderson and A. Corbett, “Cognitive tutors: Lessons learned,” *Journal of the Learning Sciences*, vol. 4, no. 2, pp. 167–207, Jan 1995.

ARTESAS, “ARTESAS - Advanced augmented reality technologies for industrial service applications.” [Online]. Available: http://www.artesas.de

R. T. Azuma, “A survey of augmented reality,” *Presence: Teleoperators and Virtual Environments*, pp. 355–385, August 1997.

K. M. Baird and W. Barfield, “Evaluating the effectiveness of augmented reality displays for a manual assembly task,” *Virtual Reality*, vol. 4, no. 4, pp. 250–259, 1999.

P. Buchanan, H. Seichter, M. Billinghurst, and R. Grasset, “Augmented reality and rigid body simulation for edutainment: the interesting mechanism - an AR puzzle to teach Newton physics,” in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology*, ser. ACE ’08. New York, NY, USA: ACM, 2008, pp. 17–20.

T. Caudell and D. Mizell, “Augmented reality: an application of heads-up display technology to manual manufacturing processes,” *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, vol. ii, pp. 659 – 669 vol.2, 1992.

D. Curtis, D. Mizell, P. Gruenbaum, and A. Janin, “Several devils in the details: making an AR application work in the airplane factory,” in *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes: placing artificial objects in real scenes*, ser. IWAR ’98. Natick, MA, USA: A. K. Peters, Ltd., 1999, pp. 47–60.

A. Dünser, “Supporting low ability readers with interactive augmented reality,” *Annual Review of CyberTherapy and Telemedicine*, Jan 2008.

M. Evers and A. Nijholt, "Jacob - An animated instruction agent in virtual reality," in *Advances in Multimodal Interfaces — ICMI 2000*, ser. Lecture Notes in Computer Science, T. Tan, Y. Shi, and W. Gao, Eds. Springer Berlin / Heidelberg, 2000, vol. 1948, pp. 526–533.

S. Feiner and S. J. Henderson, "Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret," pp. 1–10, Sep 2009.

S. Feiner, B. Macintyre, and D. Seligmann, "Knowledge-based augmented reality," *Commun. ACM*, vol. 36, pp. 53–62, July 1993.

R. Felderman and L. Silverman, "Learning and teaching styles in engineering education," *Engineering education*, Jan 1988.

M. Fjeld and B. Voegtli, "Augmented chemistry: an interactive educational workbench," *Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on*, pp. 259 – 321, 2002.

W. Friedrich, "ARVIKA - Augmented reality for development, production and service," *Proceedings of the International Symposium on Mixed and Augmented Reality*, pp. 3 – 4, 2002.

H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer, "Augmented reality visualization for laparoscopic surgery," in *Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI '98. London, UK: Springer-Verlag, 1998, pp. 934–943.

T. Furness, "The super cockpit and its human factors challenges," *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 30, no. 1, pp. 46–52, Jan 1986.

S. Gilbert, S. Blessing, and L. Blankenship, "The accidental tutor: overlaying an intelligent tutor on an existing user interface," *CHI '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, Apr 2009.

H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*, pp. 85 – 94, 1999.

K. Koedinger, J. Anderson, and W. Hadley, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30–43, 1997.

E. Lamounier, A. Bucioli, A. Cardoso, A. Andrade, and A. Soares, "On the use of augmented reality techniques in learning and interpretation of cardiologic data," *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pp. 610 – 613, 2010.

Z. Liu and Z. Pan, "An emotion model of 3d virtual characters in intelligent virtual environment," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science, J. Tao, T. Tan, and R. Picard, Eds. Springer Berlin / Heidelberg, 2005, vol. 3784, pp. 629–636.

J. Looser, R. Grasset, H. Seichter, and M. Billinghurst, "OSGART-A pragmatic approach to MR," in *5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006)*, 2006, pp. 22–25. [Online]. Available: http://technotecture.com/content/osgart-pragmatic-approach-mr

N. Major and S. Ainsworth, "REDEEM: Exploiting symbiosis between psychology and authoring environments," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 317–340, 1997.

D. McArthur and M. Lewis, "The roles of artificial intelligence in education: current progress and future prospects," *rand.org*, Jan 1993.

G. Mendez and P. Herrero, "Intelligent virtual environments for training in nuclear power plants," *Proceedings of the 6th International Conference On Enterprise Information Systems*, 2004.

A. Mitrovic, B. Martin, and P. Suraweera, "Intelligent tutors for all: Constraint-based modeling methodology, systems and authoring," *IEEE Intelligent Systems*, vol. 22, no. 4, pp. 38–45, 2007.

A. Mitrovic, N. McGuigan, B. Martin, P. Suraweera, N. Milik, and J. Holland, "Authoring constraint-based tutors in ASPIRE: A case study of a capital investment tutor," *Vienna, Austria: World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2008)*, vol. 30, pp. 4607–4616, 2008.

A. Mitrovic and S. Ohlsson, "Evaluation of a constraint-based tutor for a database language," *International Journal of Artificial Intelligence in Education*, vol. 10, pp. 238–256, 1999.

A. Munro, M. Johnson, and Q. Pizzini, "Authoring simulation-centered tutors with RIDES," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 284–316, 1997.

T. Murray, "Authoring intelligent tutoring systems: An analysis of the state of the art," *International Journal of Artificial Intelligence in Education*, Jan 1999.

W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, J. Wieghardt, H. Hörtner, and C. Lindinger, "Augmented reality navigation systems," *Universal Access in the Information Society*, vol. 4, pp. 177–187, 2006, 10.1007/s10209-005-0017-5.

U. Neumann and S. You, "Natural feature tracking for augmented reality," *Multimedia, IEEE Transactions on*, vol. 1, no. 1, pp. 53 – 64, 1999.

S. Nilsson and B. Johansson, “Fun and usable: augmented reality instructions in a hospital setting,” *OZCHI ’07: Proceedings of the 19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces*, Nov 2007.

G. Nuzzo-Jones, J. Walonoski, N. Heffernan, and T. Livak, “The extensible tutor architecture: a new foundation for ITS,” *Design of Adaptive Feedback in a Web Educational System*, p. 1, 2005.

J. Psotka and S. A. Mutter, *Intelligent Tutoring Systems: Lessons Learned.* Lawrence Erlbaum Associates, 1988.

Y. Qiao, X. Xie, and T. Sun, “Design for the cockpit intelligent tutoring system based on augmented reality,” *ISCID ’08: Proceedings of the 2008 International Symposium on Computational Intelligence and Design*, vol. 01, Oct 2008.

A. Raczynski and P. Gussmann, “Services and training through augmented reality,” *Visual Media Production, 2004. (CVMP). 1st European Conference on*, pp. 263 – 271, 2004.

D. Reiners, D. Stricker, G. Klinker, and S. Müller, “Augmented reality for construction tasks: doorlock assembly,” in *Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes: placing artificial objects in real scenes*, ser. IWAR ’98. Natick, MA, USA: A. K. Peters, Ltd., 1999, pp. 31–46.

J. Rekimoto, “Matrix: a realtime object identification and registration method for augmented reality,” *Computer Human Interaction, 1998. Proceedings. 3rd Asia Pacific*, pp. 63 – 68, 1998.

C. M. Robertson, B. MacIntyre, and B. N. Walker, “An evaluation of graphical context when the graphics are outside of the task area,” in *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR ’08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 73–76.

G. Schall, E. Mendez, E. Kruijff, E. Veas, S. Junghanns, B. Reitinger, and D. Schmalstieg, “Handheld augmented reality for underground infrastructure visualization,” *Personal Ubiquitous Computing*, vol. 13, pp. 281–291, May 2009.

R. Shelby, “An assessment of the ANDES tutor,” *oai.dtic.mil*, 2001.

B. Shelton and N. Hedley, “Using augmented reality for teaching earth-sun relationships to undergraduate geography students,” *Augmented Reality Toolkit, The First IEEE International Workshop*, p. 8 pp., 2002.

A. Tang, C. Owen, F. Biocca, and W. Mou, “Comparative effectiveness of augmented reality in object assembly,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ser. CHI ’03. New York, NY, USA: ACM, 2003, pp. 73–80.

M. Tonnis, C. Sandor, C. Lange, and H. Bubb, "Experimental evaluation of an augmented reality visualization for directing a car driver's attention," in *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 56–59.

L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3d tracking using online and offline information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 10, pp. 1385 – 1391, 2004.

S. Weghorst, "Augmenting tangible molecular models," *Proceedings of the 13th International Conference on Artificial Reality and Telexistence*, 2003.

D. Wood, J. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of Psychiatry and Psychology*, vol. 17, no. 2, pp. 89–100, 1976.

R. Wygant, "CLIPS–A powerful development and delivery expert system tool," *Computers & industrial engineering*, 1989.

J. Zauner, M. Haller, A. Brandl, and W. Hartmann, "Authoring of a mixed reality assembly instructor for hierarchical structures," in *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ser. ISMAR '03. Washington, DC, USA: IEEE Computer Society, 2003, p. 237.

Appendix A

Evaluation Documents

A.1 Human Ethics Approval

Any research study conducted at the University of Canterbury that involves human participants must first be reviewed by the Human Ethics Committee. This is to ensure that the study does not violate any ethical principles, such as the participants' rights to privacy and safety. After a study has been approved, the committee provides an official document stating their approval. The approval document issued for the user evaluation in this study is included here.

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

HUMAN ETHICS COMMITTEE

Secretary, Lynda Griffioen
Email: human-ethics@canterbury.ac.nz

Ref: HEC 2011/96/LR

28 October 2011

Giles Westerfield
Department of Computer Science & Software Engineering
UNIVERSITY OF CANTERBURY

Dear Giles

Thank you for forwarding to the Human Ethics Committee a copy of the low risk application you have recently made for your research proposal "Intelligent augmented reality for assembly and maintenance".

I am pleased to advise that this application has been reviewed and I confirm support of the Department's approval for this project.

With best wishes for your project.

Yours sincerely

Michael Grimshaw
Chair
University of Canterbury Human Ethics Committee

A.2 Consent Form

The Human Ethics Committee at the University of Canterbury requires that all study participants sign a consent form, which confirms that they agree to participate in the study and that they understand the requirements. The consent form for the user evaluation in this study is included here.

Giles Westerfield
Telephone: +64-3-364-2987
Email: giles.westerfield@canterbury.ac.nz

UNIVERSITY OF CANTERBURY
Te Whare Wānanga o Waitaha
CHRISTCHURCH NEW ZEALAND

Intelligent Augmented Reality for Assembly and Maintenance

Participant Consent Form

I have been given a full explanation of this project and have been given an opportunity to ask questions.

I understand what will be required of me if I agree to take part in this project.

I understand that my participation is voluntary and that I may withdraw at any stage without penalty.

I understand that any information or opinions I provide will be kept confidential to the researcher and that any published or reported results will not identify me.

I understand that all data collected for this study will be kept in locked and secure facilities at the University of Canterbury and will be destroyed after five years.

I understand that if I wish, I may receive a report on the findings of this study. I have provided my email details for this.

I understand that if I require further information I can contact the researcher, Giles Westerfield. If I have any complaints, I can contact Professor Tanja Mitrovic (tanja.mitrovic@canterbury.ac.nz) or the Chair of the University of Canterbury Educational Research Human Ethics Committee.

By signing below, I agree to participate in this research project.

Name: ____________________________________

Date: ____________________________________

Signature: ____________________________________

Email: ____________________________________

A.3 Information Sheet

The Human Ethics Committee at the University of Canterbury requires that all study participants be provided with an information sheet prior to signing the consent form. The information sheet describes each stage of the experiment and what is required of the participants should they agree to participate. The sheet also describes the participants' rights as well as who to contact if they have any concerns during or after the experiment. The information sheet created for the user evaluation in this study is included here.

Giles Westerfield
Telephone: +64-3-364-2987
Email: giles.westerfield@canterbury.ac.nz

Intelligent Augmented Reality for Assembly and Maintenance

Information Sheet for Participants

I am a Masters student in the Department of Computer Science and Software Engineering at the University of Canterbury. I am currently conducting a research project that investigates the combination of Augmented Reality with Intelligent Tutoring Systems, and have developed a prototype system that teaches users to assemble components on a computer motherboard.

I would like to invite you to participate in my current study. If you agree to participate, you will be asked to do the following:

- Complete a short written test to measure your prior experience with computer hardware. The test will take approximately 5 minutes.
- Follow a tutorial procedure in which you will receive visual instructions via an electronic display that is worn on your head. The instructions will guide you to identify and correctly install several different motherboard components. The tutorial process will take approximately 10 to 15 minutes.
- Complete a second written test covering the material you have learned from the tutor (5 minutes).
- Perform a hands-on test covering the material you have learned from the tutor. This will consist of identifying and installing the motherboard components once more without assistance. Data regarding your performance will be collected in order to evaluate the effectiveness of the tutor (5 minutes).
- Complete a short questionnaire regarding your experience with the tutor (5 minutes).

Please note that participation in this study is voluntary. If you do participate, you have the right to withdraw from the study at any time without penalty. If you withdraw, I will do my best to remove any information relating to you, provided this is practically achievable.

I will take particular care to ensure the confidentiality of all data gathered for this study. I will also take care to ensure your anonymity in publications of the findings. All the data will be securely stored in password protected facilities and locked storage at the University of Canterbury for five years following the study. It will then be destroyed.

The results of this research may be used to improve the design of Augmented Reality-based tutors. The results may also be reported internationally at conferences and in scientific journals. You may request a report describing the findings by indicating your preference on the consent form.

Please contact me if you have any questions about the study (details above). If you have a complaint about the study, you may contact the Chair, Educational Research Human Ethics Committee, University of Canterbury, Private Bag 4800, Christchurch (human-ethics@canterbury.ac.nz).

If you agree to participate in this study, please complete the attached consent form and return it to me in person or by mail to HITLabNZ, University of Canterbury, Private Bag 4800, Christchurch 8140.

I am looking forward to working with you and thank you in advance for your contributions.

Giles Westerfield

A.4 Written Tests

The evaluation participants were given two written tests during the experiment. The first was a pre-test designed to evaluate their existing computer hardware knowledge prior to using either the intelligent or traditional AR tutor. The second test was a post-test designed to evaluate the participants' computer hardware knowledge after using one of the AR tutors. Both of the tests are included here.

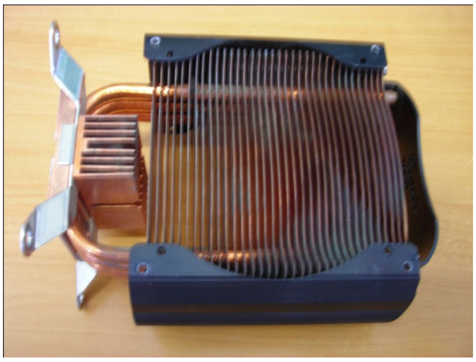
Subject ID: _______

Intelligent Augmented Reality for Assembly and Maintenance

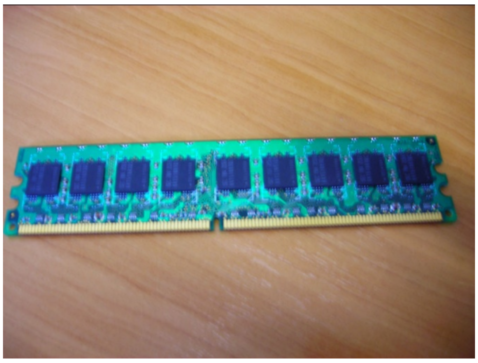
Written Pre-Test

The purpose of this test is to measure your computer hardware knowledge before using the tutor.

1. Please name the computer hardware component in each picture.



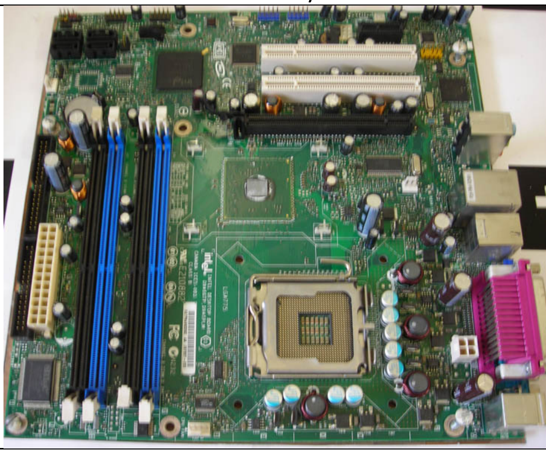
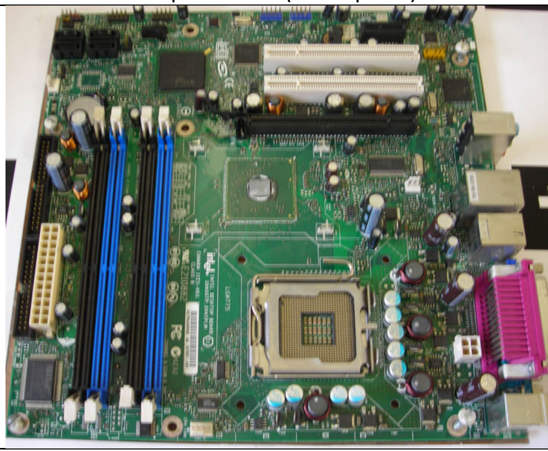
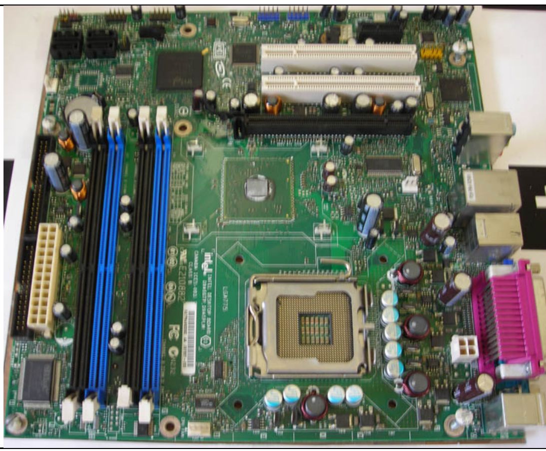
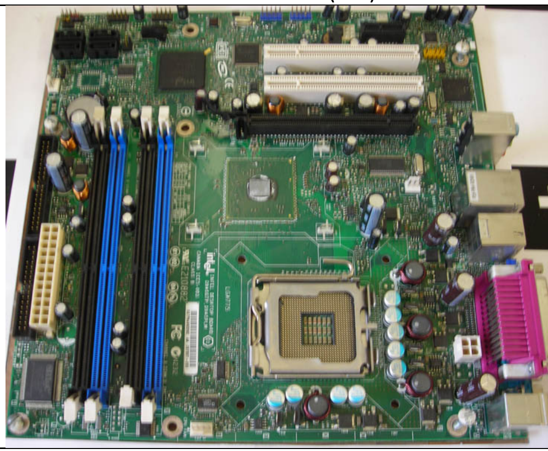
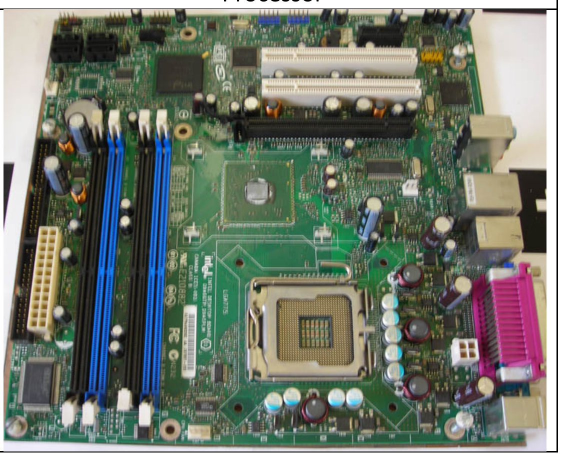
___________________________________ ___________________________________



___________________________________ ___________________________________

____

2. For each component, circle a valid location on the motherboard where it can be installed.

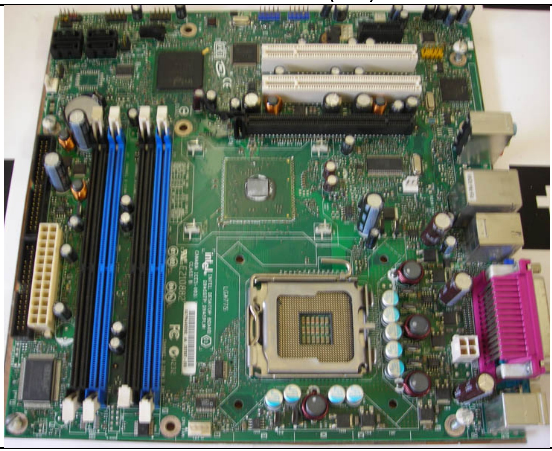
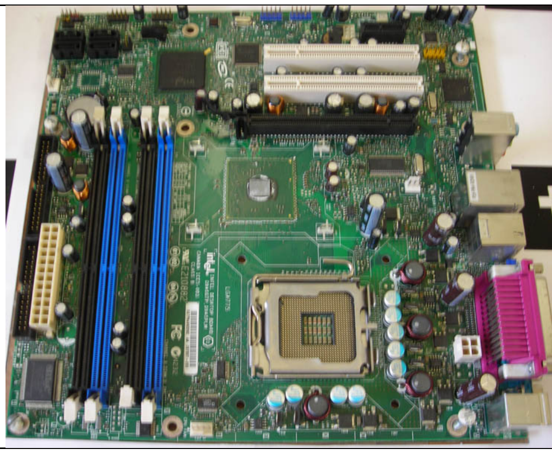
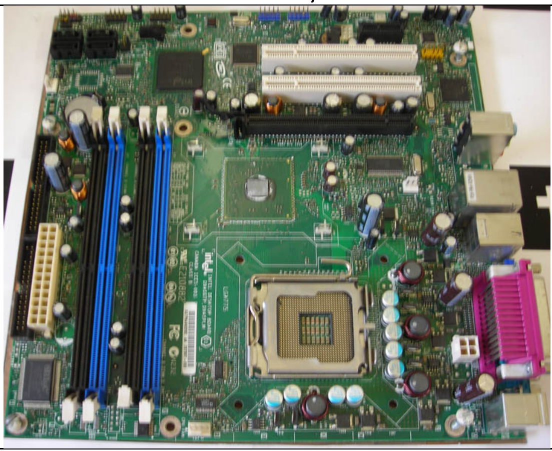
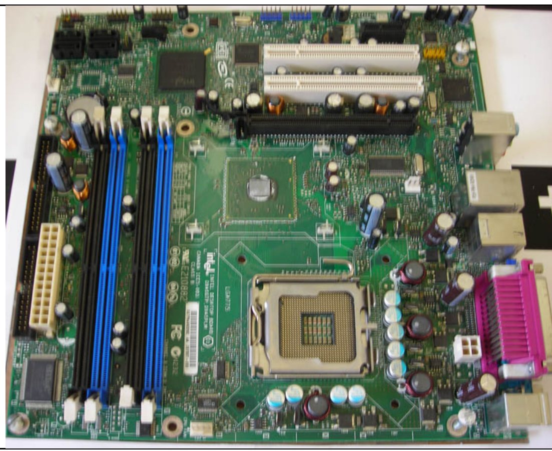
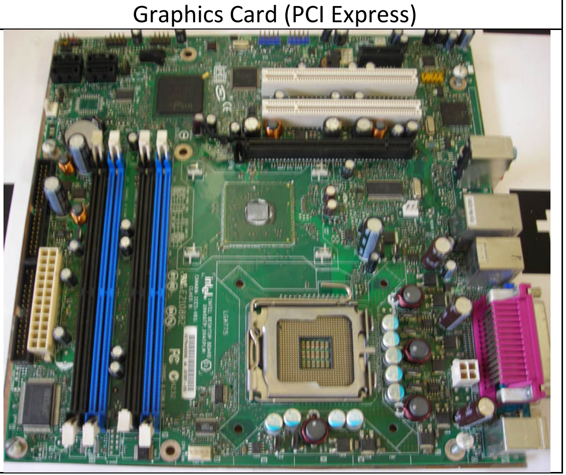
Memory	Graphics Card (PCI Express)
	
Heatsink	TV Tuner Card (PCI)
	
Processor	
	

Subject ID: _______

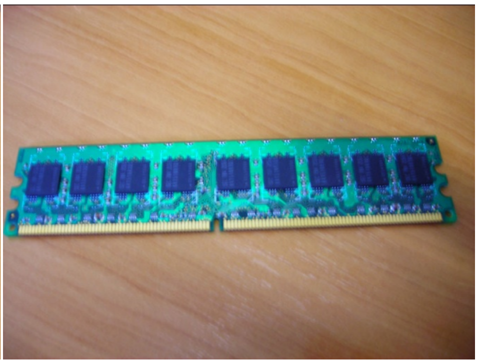
Intelligent Augmented Reality for Assembly and Maintenance

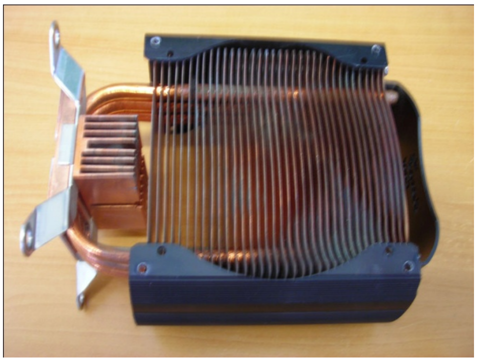
Written Post-Test

1. For each component, circle a valid location on the motherboard where it can be installed.

TV Tuner Card (PCI)	Processor
	
Memory	Heatsink
	
Graphics Card (PCI Express)	
	

2. Please name the computer hardware component in each picture.





A.5 Questionnaire

At the end of the experiment, the participants were given a questionnaire containing a series of Likert-scale ratings and open-ended questions designed to capture their subjective opinions regarding the AR tutoring experience. The questionnaire is included here.

Intelligent Augmented Reality for Assembly and Maintenance

Questionnaire

Thank you for participating in my evaluation study—your feedback is crucial to my current and future research. This questionnaire is anonymous and if you wish, you may at any time withdraw from participation, including withdrawal of any information you have provided. However, by completing the questionnaire, you indicate your consent for publication of the generalized results of our research findings.

1. Please indicate your level of experience with computer hardware assembly procedures **prior to participating in this study**.

Not very experienced						Very experienced
1	2	3	4	5	6	7

2. Was the augmented reality tutor able to teach you how to correctly install the motherboard components?

Not very much						Very much
1	2	3	4	5	6	7

3. Do you think it was easy to perform the required tasks during the tutoring process?

Not very much						Very much
1	2	3	4	5	6	7

4. Do you think the augmented reality tutor was effective at teaching the installation procedures for various computer components?

Not very much						Very much
1	2	3	4	5	6	7

5. Please share any additional feedback you may have regarding the overall effectiveness of the tutor.

6. Were you satisfied with the level of feedback (graphics and text) you received from the tutor as you performed the tasks?

Not very much						Very much
1	2	3	4	5	6	7

7. Please provide any additional feedback you may have regarding the graphical and textual feedback.

8. Do you think the augmented reality tutor was a more effective method of teaching motherboard installation procedures than printed instructions (e.g. reading a manual)?

Not very much						Very much
1	2	3	4	5	6	7

9. Do you think the augmented reality tutor was a more effective method of teaching the motherboard installation procedures than an instructional video?

Not very much						Very much
1	2	3	4	5	6	7

10. Do you think the augmented reality tutor is interesting to use?

Not very much						Very much
1	2	3	4	5	6	7

11. Please indicate the general level of frustration or mental stress you felt while using the tutor.

Not very stressed						Very stressed
1	2	3	4	5	6	7

12. Please indicate the general level of physical stress you felt while using the tutor.

Not very stressed						Very stressed
1	2	3	4	5	6	7

13. Please provide any additional comments about your experience. This could include details about what you liked, what you didn't like, or what could be improved.